

EXHIBIT A



US010970588B2

(12) **United States Patent**
Harron et al.

(10) **Patent No.:** **US 10,970,588 B2**

(45) **Date of Patent:** ***Apr. 6, 2021**

(54) **RECURRENT DEEP NEURAL NETWORK SYSTEM FOR DETECTING OVERLAYS IN IMAGES**

(71) Applicant: **Gracernote, Inc.**, Emeryville, CA (US)

(72) Inventors: **Wilson Harron**, Berkeley, CA (US);
Irene Zhu, San Francisco, CA (US)

(73) Assignee: **Gracernote, Inc.**, Emeryville, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **16/699,536**

(22) Filed: **Nov. 29, 2019**

(65) **Prior Publication Data**

US 2020/0104625 A1 Apr. 2, 2020

Related U.S. Application Data

(63) Continuation of application No. 15/793,352, filed on Oct. 25, 2017, now Pat. No. 10,503,998.

(Continued)

(51) **Int. Cl.**
G06K 9/00 (2006.01)
G06K 9/46 (2006.01)

(Continued)

(52) **U.S. Cl.**
CPC **G06K 9/4609** (2013.01); **G06K 9/34** (2013.01); **G06K 9/344** (2013.01); **G06K 9/4628** (2013.01);

(Continued)

(58) **Field of Classification Search**

CPC G06K 9/4609; G06K 9/34; G06K 9/344;
G06K 9/4628; G06K 2209/01;

(Continued)

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,341,142 A * 8/1994 Reis F41G 7/2226
244/3.15

6,578,021 B1 * 6/2003 Barillaud H04L 12/4641
706/20

(Continued)

OTHER PUBLICATIONS

Gers, Felix et al., Recurrent Nets that Time and Count, IDSIA, Lugano, Switzerland, Jul. 2000 (6 pages).

(Continued)

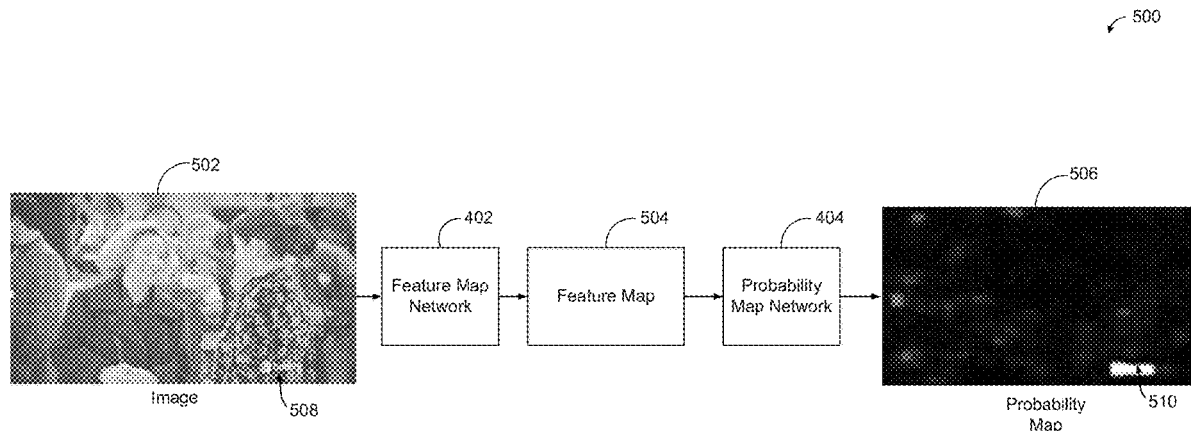
Primary Examiner — Amir Alavi

(74) *Attorney, Agent, or Firm* — McDonnell Boehnen Hulbert & Berghoff LLP

(57) **ABSTRACT**

In one aspect, an example method includes a processor (1) applying a feature map network to an image to create a feature map comprising a grid of vectors characterizing at least one feature in the image and (2) applying a probability map network to the feature map to create a probability map assigning a probability to the at least one feature in the image, where the assigned probability corresponds to a likelihood that the at least one feature is an overlay. The method further includes the processor determining that the probability exceeds a threshold, and responsive to the processor determining that the probability exceeds the threshold, performing a processing action associated with the at least one feature.

20 Claims, 12 Drawing Sheets



US 10,970,588 B2

Page 2

Related U.S. Application Data					
		9,767,381	B2 *	9/2017	Rodriguez-Serrano G06F 16/54
(60)	Provisional application No. 62/421,639, filed on Nov. 14, 2016, provisional application No. 62/418,517, filed on Nov. 7, 2016.	9,846,840	B1 *	12/2017	Lin G06F 16/5854
		10,354,168	B2	7/2019	Bluche
		10,503,998	B2 *	12/2019	Harron G06N 3/0481
		10,572,725	B1 *	2/2020	Becker G06K 9/00442
		10,699,135	B2 *	6/2020	Zhang G08G 1/167
(51)	Int. Cl.	2004/0076984	A1 *	4/2004	Eils G16B 40/00
	G06N 3/04 (2006.01)				435/6.13
	G06K 9/34 (2006.01)	2010/0124364	A1	5/2010	Huo et al.
	G06N 3/08 (2006.01)	2018/0005379	A1	1/2018	Den Hartog et al.
(52)	U.S. Cl.	2018/0129899	A1	5/2018	Harron et al.
	CPC G06N 3/0445 (2013.01); G06N 3/0454 (2013.01); G06N 3/0472 (2013.01); G06N 3/0481 (2013.01); G06N 3/08 (2013.01); G06K 2209/01 (2013.01)	2018/0336454	A1	11/2018	Lim et al.
(58)	Field of Classification Search	OTHER PUBLICATIONS			
	CPC .. G06N 3/0445; G06N 3/0454; G06N 3/0472; G06N 3/0481; G06N 3/08; G06T 7/277	Hochreiter, Sepp et al., Long Short-Term Memory, Neural Computation 9(8):1735-1780, 1997 (32 pages).			
	See application file for complete search history.	Huiskes, Mark et al., The MIRFLICKR Retrieval Evaluation (image collection, complete manual annotations and open software), downloaded Oct. 25, 2017 from the world-wide web at http://press.liacs.nl/mirflickr/ (5 pages).			
(56)	References Cited	Huval, Brody et al., An Empirical Evaluation of Deep Learning on Highway Driving, arXiv, Apr. 2015 (7 pages).			
	U.S. PATENT DOCUMENTS	Schuster, Mike et al., Bidirectional Recurrent Neural Networks, IEEE Transactions on Signal Processing, vol. 45, No. 11, Nov. 1997, pp. 2673-2681 (9 pages).			
	8,135,199 B2 * 3/2012 Russakoff G06K 9/6209	Sermanet, Pierre et al., OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks, arXiv, Feb. 2014 (16 pages).			
		Simonyan, Karen et al., Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR, arXiv, Apr. 2015 (14 pages).			
	8,380,384 B2 * 2/2013 Lee G06T 7/246				
	8,436,913 B2 5/2013 Lanz				
	8,818,099 B2 * 8/2014 Tian G06K 9/38				
	9,514,363 B2 * 12/2016 Sigal G06K 9/6263				

* cited by examiner

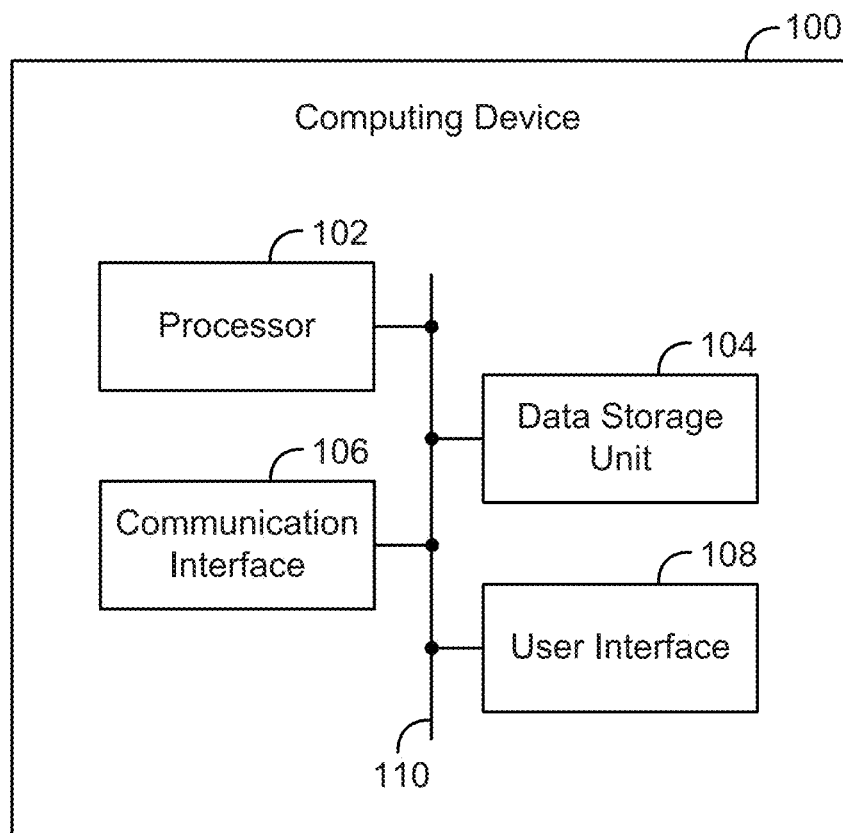


Figure 1

U.S. Patent

Apr. 6, 2021

Sheet 2 of 12

US 10,970,588 B2

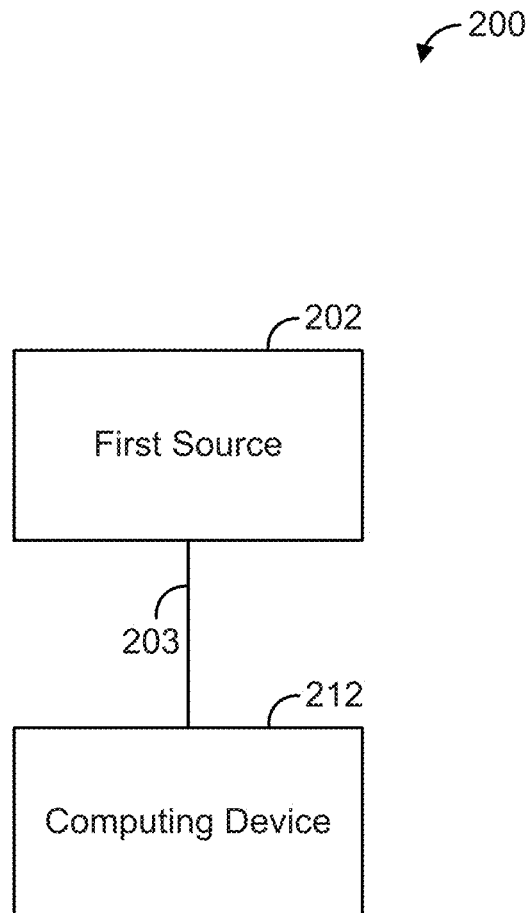


Figure 2

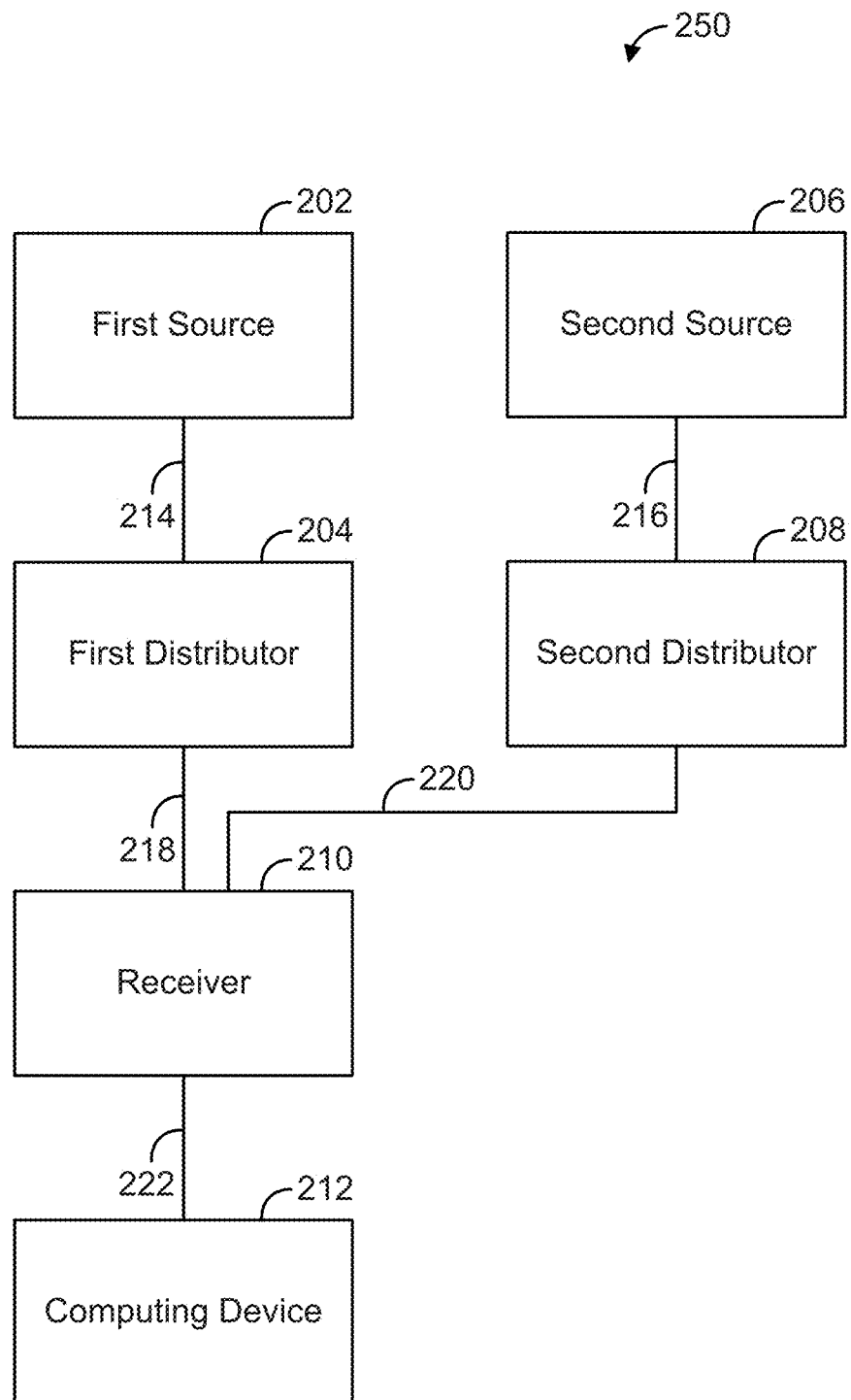


Figure 3

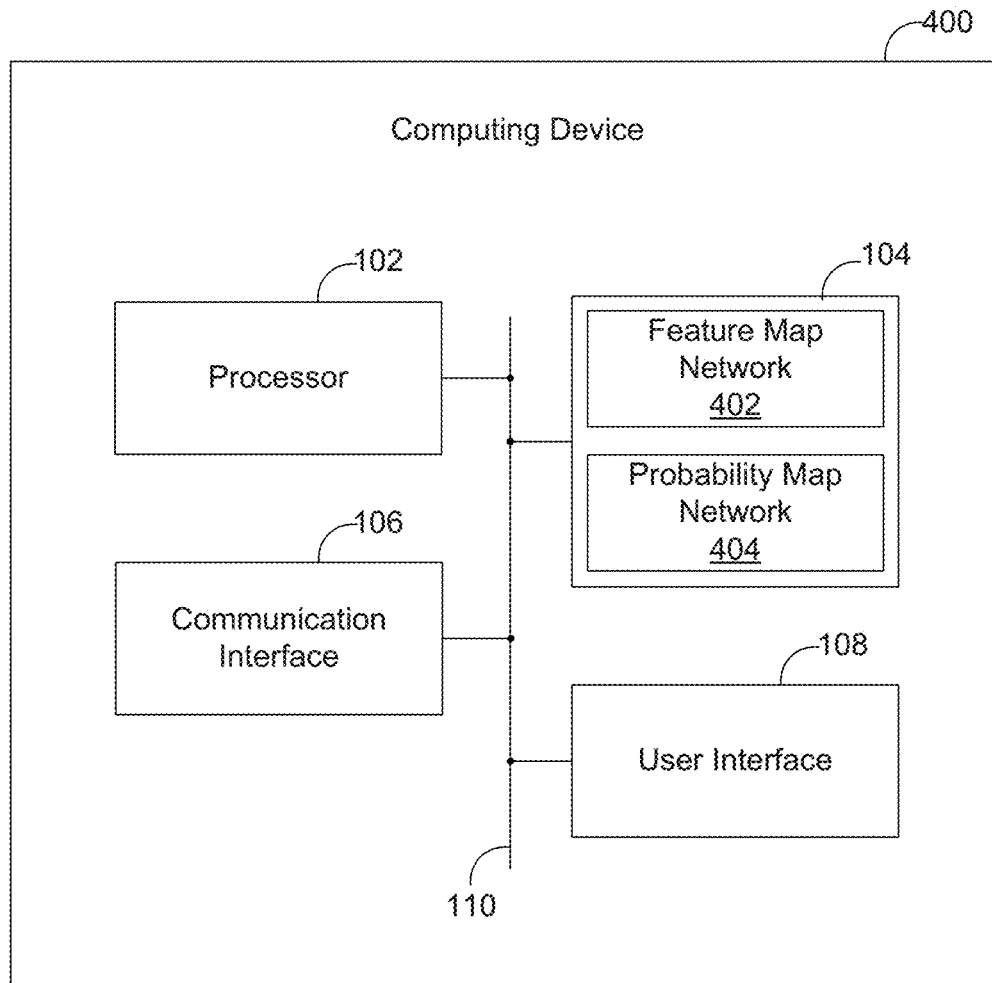


Figure 4

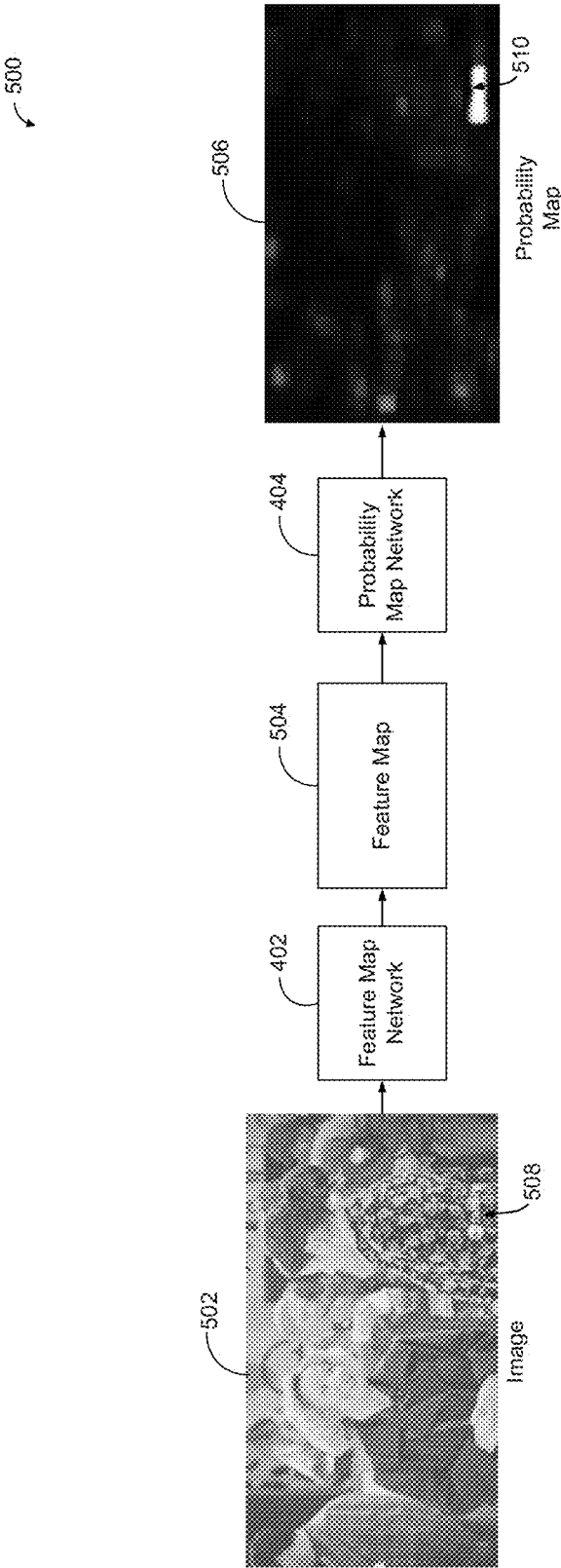


Figure 5

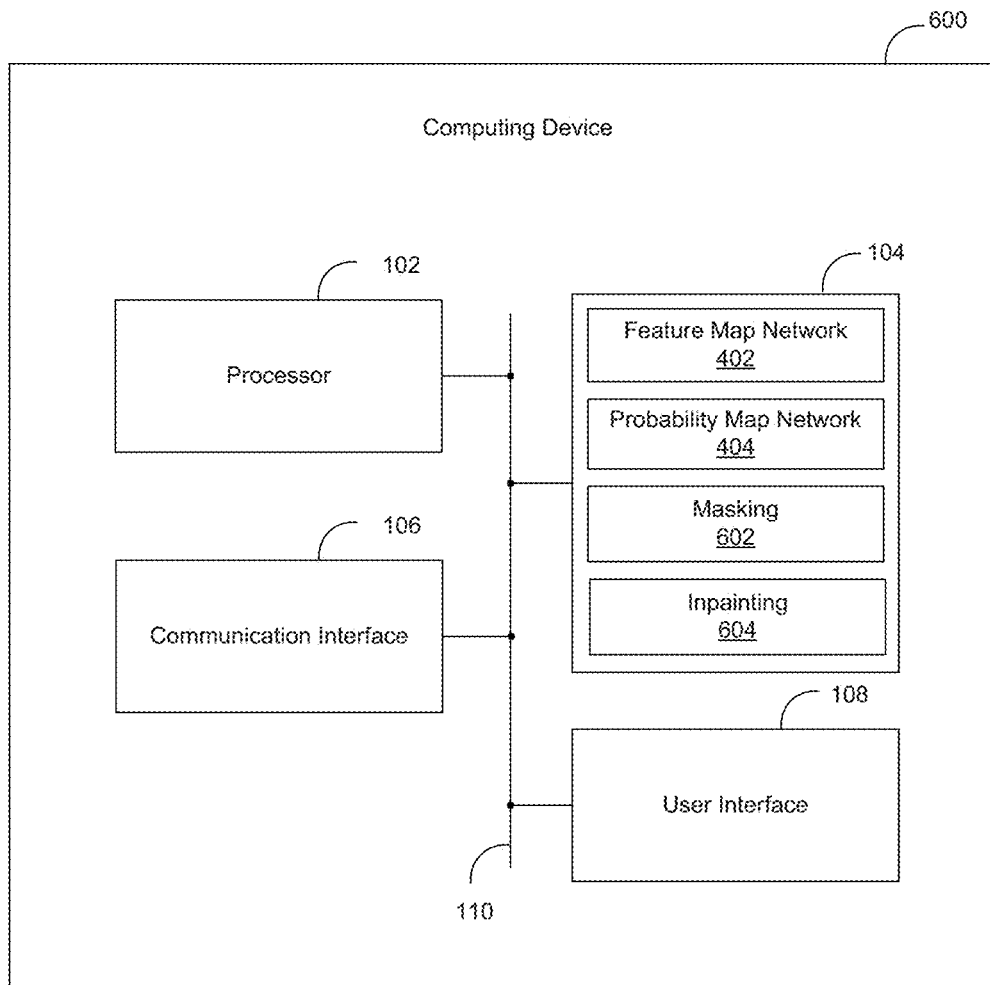


Figure 6

U.S. Patent

Apr. 6, 2021

Sheet 7 of 12

US 10,970,588 B2

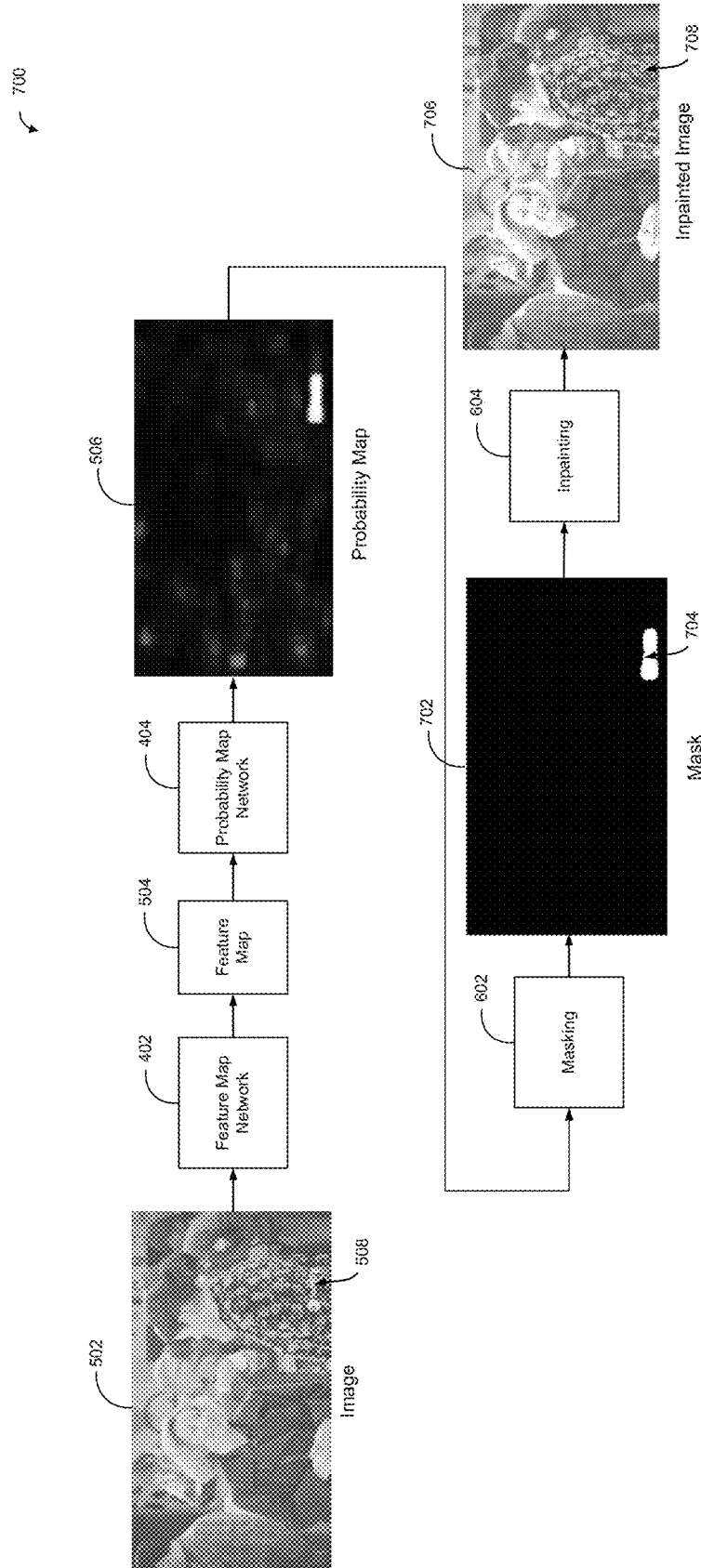


Figure 7

U.S. Patent

Apr. 6, 2021

Sheet 8 of 12

US 10,970,588 B2

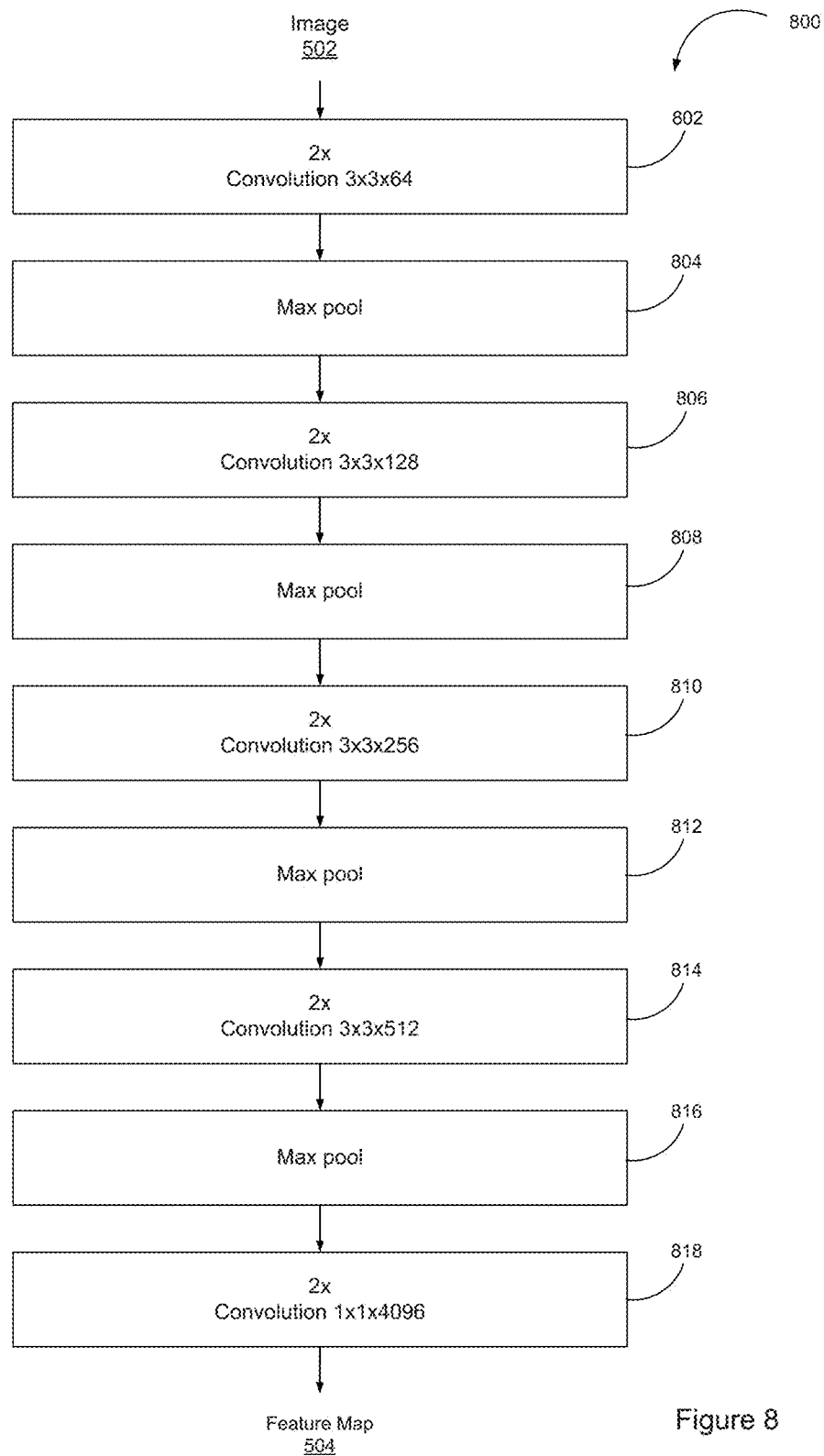


Figure 8

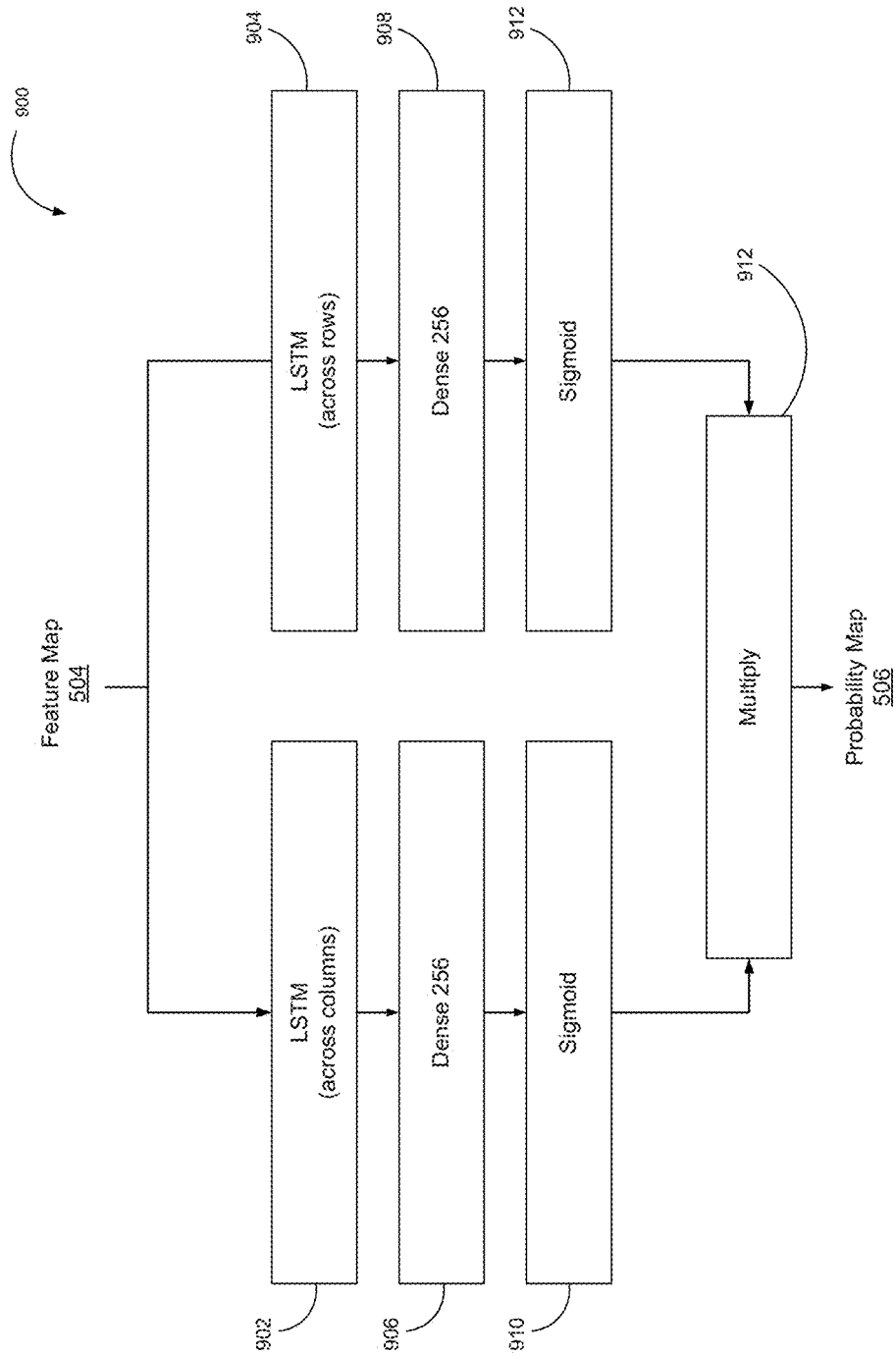


Figure 9

U.S. Patent

Apr. 6, 2021

Sheet 10 of 12

US 10,970,588 B2

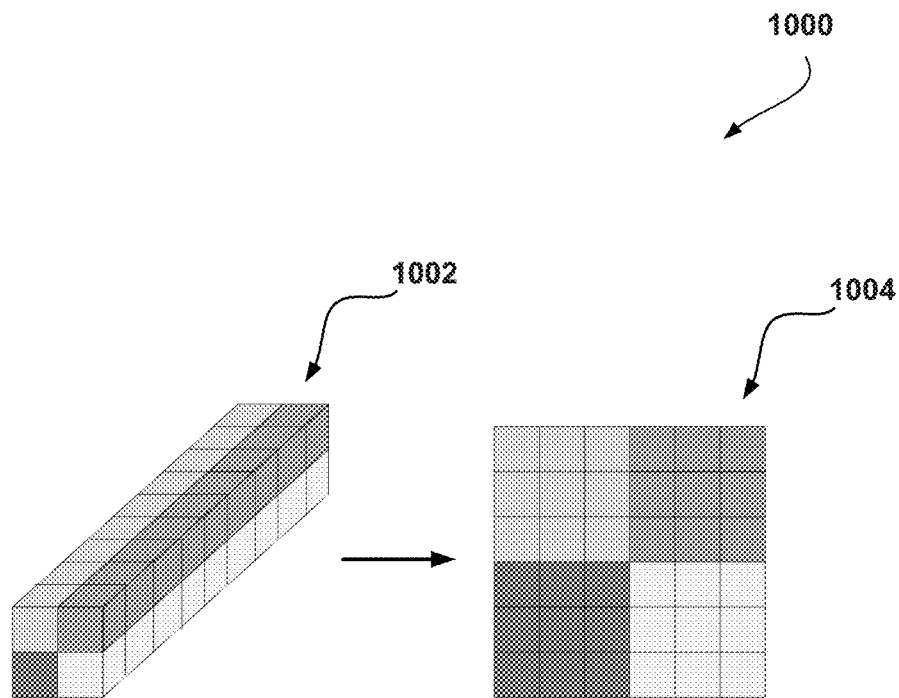


Figure 10

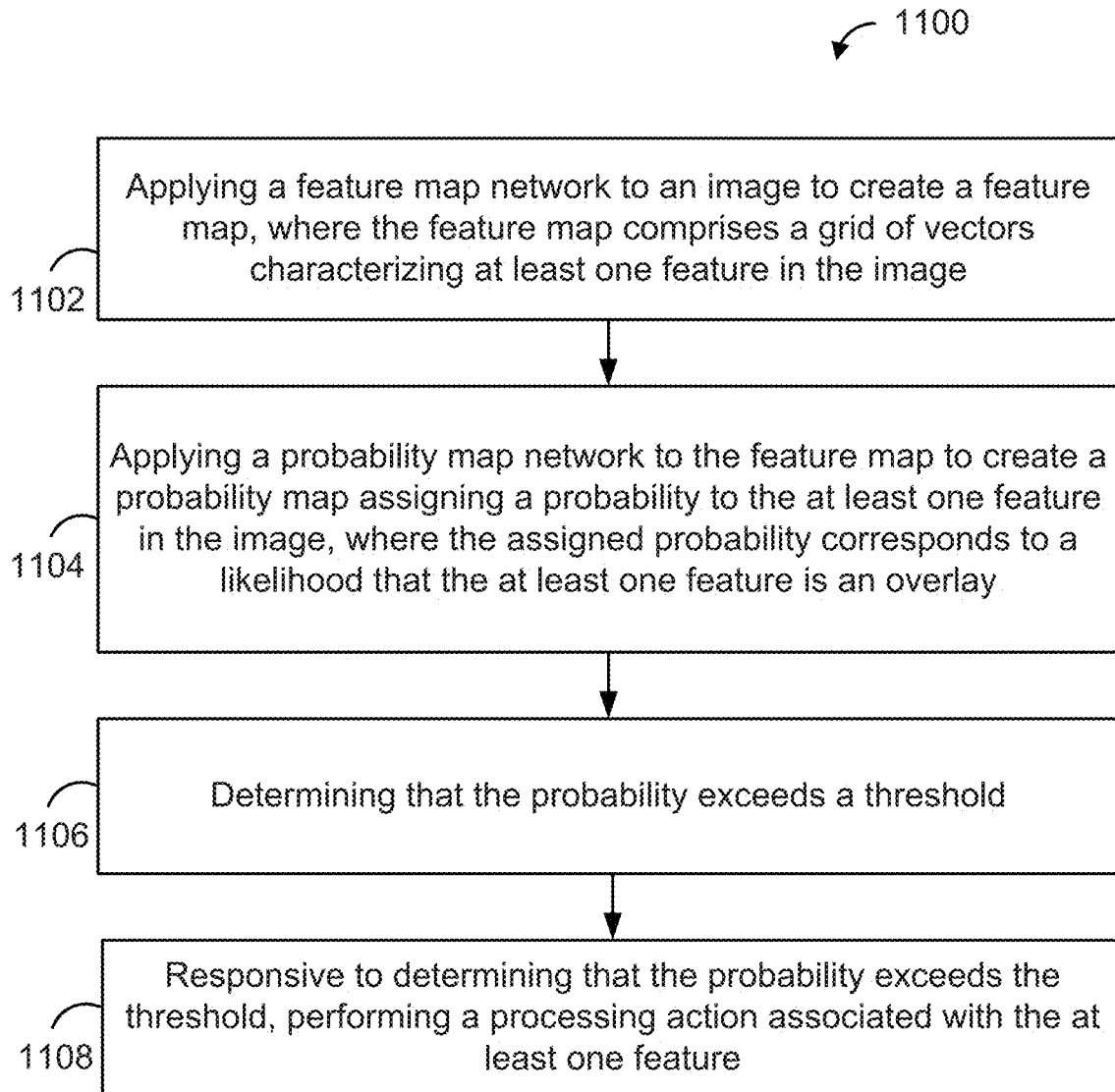


Figure 11

U.S. Patent

Apr. 6, 2021

Sheet 12 of 12

US 10,970,588 B2

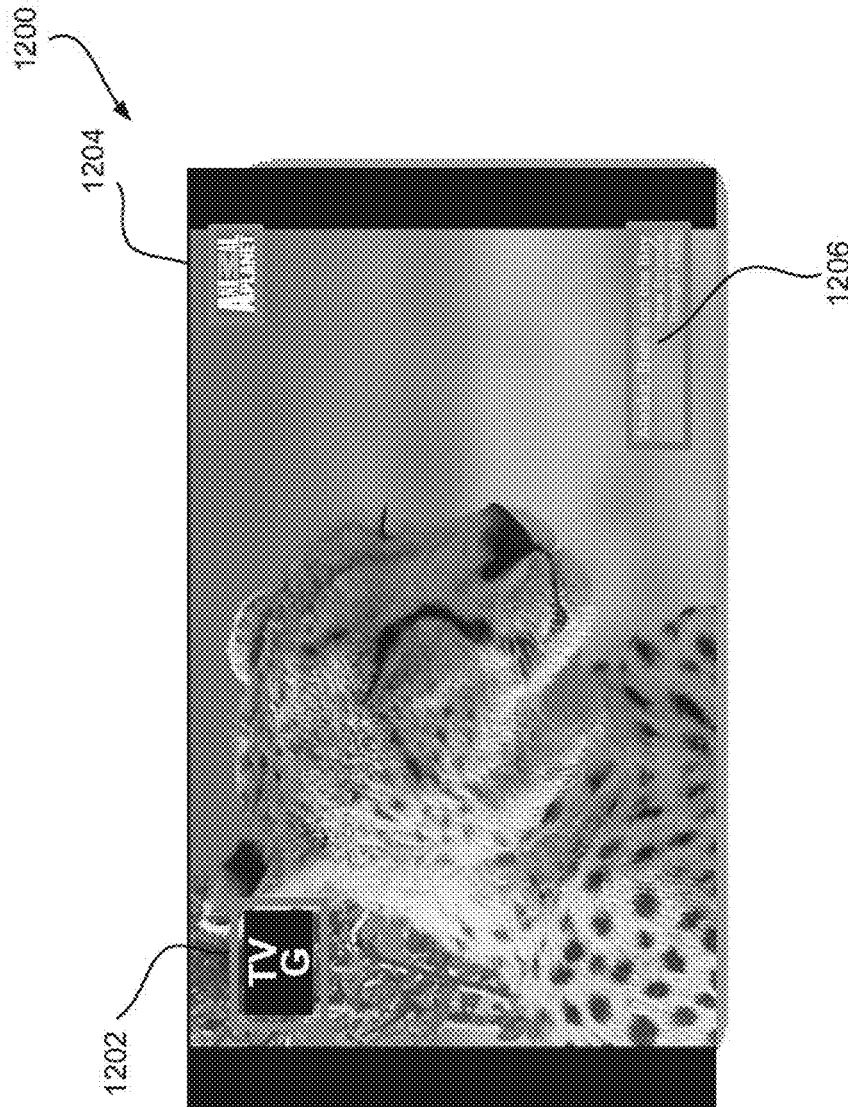


Figure 12

US 10,970,588 B2

1

RECURRENT DEEP NEURAL NETWORK SYSTEM FOR DETECTING OVERLAYS IN IMAGES

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation of U.S. patent application Ser. No. 15/793,352, filed Oct. 25, 2017, which claims priority to U.S. Provisional Patent Application No. 62/418,517, filed Nov. 7, 2016, and U.S. Provisional Patent Application No. 62/421,639, filed Nov. 14, 2016. All of the aforementioned applications are incorporated herein by reference in their entireties.

USAGE AND TERMINOLOGY

In this disclosure, unless otherwise specified and/or unless the particular context clearly dictates otherwise, the terms “a” or “an” mean at least one, and the term “the” means the at least one.

SUMMARY

In one aspect, an example method is disclosed. The method includes a processor (1) applying a feature map network to an image to create a feature map comprising a grid of vectors characterizing at least one feature in the image and (2) applying a probability map network to the feature map to create a probability map assigning a probability to the at least one feature in the image, where the assigned probability corresponds to a likelihood that the at least one feature is an overlay. The method further includes the processor determining that the probability exceeds a threshold. Upon determining that the probability exceeds the threshold, the method includes performing a processing action associated with the at least one feature.

In another aspect, an example non-transitory computer-readable medium is disclosed. The computer-readable medium has stored thereon program instructions that upon execution by a processor, cause performance of a set of acts including applying a feature map network to an image to create a feature map comprising a grid of vectors characterizing a feature in the image, applying a probability map network to the feature map to create a probability map assigning a probability to the feature in the image, where the assigned probability corresponds to a likelihood that the feature is an overlay, determining that the probability exceeds a threshold, and upon determining that the probability exceeds the threshold, and performing a processing action associated with the feature.

In yet another aspect, a system is disclosed. The system includes a processor configured to execute program instructions to cause functions to be performed. Those functions include training a feature map network by generating a plurality of groundtruth masks from a corresponding plurality of training images, where each of the plurality of training images has an overlay with at least one of a respective grayscale intensity, color hue, color value, color intensity, opacity, size, or position that differs relative to other overlays in the plurality of training images. Another function performed is applying the trained feature map network to an image to create a feature map, where the feature map includes data characterizing a feature in the image relative to other features in the image, where the feature map network comprises a very deep convolutional neural network having a plurality of convolutional layers

2

and max-pooling layers. Yet another function performed is applying a probability map network to the feature map to create a probability map, where the probability map network comprises at least two recurrent layers including at least one vertical recurrent layer and at least one horizontal recurrent layer that are multiplied to provide an output indicative of a probability that the feature in the image is an overlay. Still yet another function performed is determining that the probability exceeds a threshold. Upon determining that the probability exceeds the threshold, a processing action associated with the feature is performed, where the processing action is selected from the group consisting of removing the feature from the image and inpainting the feature.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a simplified block diagram of an example computing device.

FIG. 2 is a simplified block diagram of an example system for detecting overlays,

FIG. 3 is a simplified block diagram of another example system for detecting overlays.

FIG. 4 is a simplified block diagram of a computing device configured to detect overlays in one or more images.

FIG. 5 is a simplified block diagram illustrating a system for detecting an overlay in an image.

FIG. 6 is a simplified block diagram of a computing device configured to detect an overlay and inpaint the detected overlay.

FIG. 7 is a simplified block diagram illustrating a system for detecting an overlay in an image and inpainting the detected overlay.

FIG. 8 is a simplified block diagram of an example feature map network.

FIG. 9 is a simplified block diagram of an example probability map network.

FIG. 10 is a simplified block diagram of illustrating a raveling operation, in which probability vectors are raveled into respective patch positions in a probability map.

FIG. 11 is a flow chart illustrating an example method for detecting an overlay in an image.

FIG. 12 is an example image with overlays.

DETAILED DESCRIPTION

I. Overview

Embodiments of the systems and methods described herein can be used to detect semitransparent or opaque overlays, such as logos and/or text, in images, such as stills captured from TV shows or other media presentations. For example, FIG. 12 is a simplified diagram of an image 1200 with overlays, including an opaque rating overlay 1202, a semitransparent station identifier overlay 1204, and a semitransparent advertising overlay 1206. A detector that is invariant to size, color, and opacity is designed to locate overlays commonly found in broadcast TV streams, for example. In some embodiments, the system uses a deep recurrent neural network to produce a probability map of the areas where one or more overlays are likely to be present in the image. Using this probability map, a mask can be made to extract the one or more of the overlays for further processing action(s). Such further processing actions could include additional inspection, removal (e.g. using an inpainting algorithm), channel recognition (e.g. using Optical Character Recognition (OCR)), or tracking, such as for determining compliance with advertising contracts, for example,

US 10,970,588 B2

3

in which compensation is based on whether an overlay is present in a particular image or sequence of images.

In conventional logo removal techniques, a human editor is required to manually view and mark unwanted regions (e.g. an overlay) in each image (or series of images) to create a mask with an overlay location, to which a conventional inpainting algorithm can be applied to create a modified image with the overlay removed. Unlike in these conventional logo removal techniques, the systems and methods described herein utilize machine learning principles to determine which areas are likely to contain an overlay. For example, the utilized machine learning principles include one or more aspects relating to neural networks, such as recurrent deep neural networks utilizing a Long Short-Term Memory (LSTM) architecture in two directions (e.g. horizontal and vertical) in an image.

As a result the technology described herein provides an improvement to the technical field of image processing, and, in particular, to improving speed and reliability of detecting overlays in images, with reduced or eliminated human oversight.

II. Example Architecture

A. Computing Device

FIG. 1 is a simplified block diagram of an example computing device 100. The computing device 100 can be configured to perform and/or can perform one or more acts and/or functions, such as those described in this disclosure. The computing device 100 can include various components, such as a processor 102, a data storage unit 104, a communication interface 106, and/or a user interface 108. Each of these components can be connected to each other via a connection mechanism 110.

In this disclosure, the term “connection mechanism” means a mechanism that facilitates communication between two or more components, devices, systems, or other entities. A connection mechanism can be a relatively simple mechanism, such as a cable or system bus, or a relatively complex mechanism, such as a packet-based communication network (e.g., the Internet). In some instances, a connection mechanism can include a non-tangible medium (e.g., in the case where the connection is wireless).

The processor 102 can include a general-purpose processor (e.g., a microprocessor) and/or a special-purpose processor (e.g., a digital signal processor (DSP)). The processor 102 can execute program instructions included in the data storage unit 104 as discussed below.

The data storage unit 104 can include one or more volatile, non-volatile, removable, and/or non-removable storage components, such as magnetic, optical, and/or flash storage, and/or can be integrated in whole or in part with the processor 102. Further, the data storage unit 104 can take the form of a non-transitory computer-readable storage medium, having stored thereon program instructions (e.g., compiled or non-compiled program logic and/or machine code) that, upon execution by the processor 102, cause the computing device 100 to perform one or more acts and/or functions, such as those described in this disclosure. These program instructions can define, and/or be part of, a discrete software application. In some instances, the computing device 100 can execute program instructions in response to receiving an input, such as an input received via the communication interface 106 and/or the user interface 108. The data storage unit 104 can also store other types of data, such as those types described in this disclosure.

4

The communication interface 106 can allow the computing device 100 to connect with and/or communicate with another entity according to one or more protocols. In one example, the communication interface 106 can be a wired interface, such as an Ethernet, High-Definition Multimedia interface (MIDI), or coaxial interface. In another example, the communication interface 106 can be a wireless interface, such as a cellular or WI-FI interface. In this disclosure, a connection can be a direct connection or an indirect connection, the latter being a connection that passes through and/or traverses one or more entities, such as a router, switcher, or other network device. Likewise, in this disclosure, a transmission can be a direct transmission or an indirect transmission.

The user interface 108 can include hardware and/or software components that facilitate interaction between the computing device 100 and a user of the computing device 100, if applicable. As such, the user interface 108 can include input components such as a keyboard, a keypad, a mouse, a remote control, a touch-sensitive panel, and/or a microphone, and/or output components such as a display device (which, for example, can be combined with a touch-sensitive panel), a sound speaker, and/or a haptic feedback system.

The computing device 100 can take various forms, such as a server, a workstation terminal, a desktop computer, a laptop, a tablet, a mobile phone, a set-top box receiver, and/or a TV.

B. Example Systems

FIG. 2 is a simplified block diagram of an example system 200. The system 200 can perform various acts and/or functions related to detecting overlays in images, and can be implemented as a computing system. In this disclosure, the term “computing system” means a system that includes at least one computing device. In some instances, a computing system can include one or more other computing systems.

The system 200 can include various components, such as a first image source (referred to herein as a “first source”) 202 and a computing device 212, both of which can be implemented as a computing system. The system 200 can also include a connection mechanism 203, which connects the first source 202 with the computing device 212.

The first source 202 can take various forms. For example, the first source 202 can be a broadcaster and/or a web server. In some cases, the first source 202 can be integrated together with a distributor (not shown), such as a multi-channel distributor (e.g., a local content provider, a cable provider, a satellite provider, an over-the-air broadcast provider, and/or a web aggregator) and/or an Internet service provider. As another alternative, the first source 202 can be a database, such as a local image database stored locally on a computer-readable medium and/or a remote image database administered by one or more servers.

The computing device 212 can each take various forms as well. For example, the computing device 212 can take the form of the computing device 100 illustrated in FIG. 1. The computing device 212 can include a cable receiver, a satellite receiver, an over-the-air broadcast receiver, and/or a streaming media receiver to receive one or more images, such as in an image stream. The computing device 212 can also include a user interface, including an input device and/or an output device. For example, the input device could include a mouse, keyboard, touchscreen or other input mechanism, while the output device could include a TV or another display device, a loudspeaker or another audio device, and/or a printing device.

US 10,970,588 B2

5

FIG. 3 is a simplified block diagram of a system **250**. Like the system **200**, the system **250** can perform various acts and/or functions related to detecting overlays in images, and can be implemented as a computing system.

The system **250** can include various components, such as the first image source (referred to herein as the “first source”) **202** (see FIG. 2), a first image distributor (referred to herein as a “first distributor”) **204**, a second image source (referred to herein as a “second source”) **206**, a second image distributor (referred to herein as a “second distributor”) **208**, an image receiver (referred to herein as a “receiver”) **210**, and/or the computing device **212** (see FIG. 2), each of which can be implemented as a computing system.

The system **250** can also include a connection mechanism **214**, which connects the first source **202** with the first distributor **204**; a connection mechanism **216**, which connects the second source **206** with the second distributor **208**; a connection mechanism **218**, which connects the first distributor **204** with the receiver **210**; a connection mechanism **220**, which connects the second distributor **208** with the receiver **210**; and a connection mechanism which connects the receiver **210** with the computing device **212**. In practice, the system **250** is likely to include many of some or all of the example components described above, which can allow for many image sources to provide media content to many computing devices or other types of devices, such as display devices (e.g. TVs).

The first source **202** and the first distributor **204** can each take various forms. For example, the first source **202** can be a broadcaster and/or a web server, and the first distributor **204** can be a multi-channel distributor (e.g., a local content provider, a cable provider, a satellite provider, an over-the-air broadcast provider, and/or a web aggregator) and/or an Internet service provider. Likewise, the second source **206** and the second distributor **208** can take various forms such as the respective examples provided above. In some cases, the first source **202** and the first distributor **204** can be integrated together. Likewise, the second source **206** and the second distributor **208** can be integrated together.

The receiver **210** and the computing device **212** can each take various forms as well. For example, the receiver can be a cable receiver, a satellite receiver, an over-the-air broadcast receiver, and/or a streaming media receiver, and the computing device **212** can be a TV or another display device, a loudspeaker or another audio device, or a printing device. In some cases, the receiver **210** and the computing device **212** can be integrated together.

FIG. 4 is a simplified block diagram of a computing device **400** configured to detect overlays in one or more images. The computing device **400** includes the components of the computing device **100** illustrated in FIG. 1 and also includes two additional components: a feature map network **402** and a probability map network **404**. The feature map network **402** receives as an input at least one image, such as a plurality of images in an image stream (e.g. video dip), and outputs a feature map, which may be in the form of a grid of feature vectors each having a particular length. The probability map network **404** receives as an input the feature map and learns (via one or more machine-learning techniques, such as those disclosed herein) a map of probabilities ranging from 0 to 1 (with 0 being least probable and 1 being most probable) corresponding to the likelihood that particular pixels (or groups of pixels) are part of an overlay.

Both the feature map network **402** and probability map network **404** in the example of FIG. 4 are neural networks implemented on one or more computing systems that

6

include one or more on-board, off-board, or remote processors, memories, interfaces, and corresponding communicative connections to implement the methods and systems described herein. The one or more computing systems can take various forms, such as a workstation terminal, a desktop computer, or a laptop, for example. In accordance with the overlay detection systems and methods disclosed herein, the one or more computing systems may access a non-transitory computer-readable medium that has program instructions stored thereon. Upon execution by a processor, the program instructions cause performance of a set of acts that includes applying the feature map network **402** to an image to create a feature map that has data characterizing a feature (e.g. a graphical object) in the image relative to other features in the image. The set of acts further includes applying the probability map network **404** to the feature map to create a probability map assigning a probability to the feature in the image. The assigned probability may be a likelihood that the feature is an overlay, for example. Upon determining that the feature is probably an overlay (i.e. the probability exceeds a threshold), a processing action can be performed on the feature, such as removing the feature (e.g. removing the overlay using an inpainting algorithm), using OCR on the feature to identify a channel or show described by the associated overlay, or tracking, such as for determining compliance with advertising contracts, for example, in which compensation is based on whether an overlay is present in a particular image or sequence of images. Other processing actions may be performed on the feature.

FIG. 5 is a simplified block diagram illustrating a system **500** for detecting an overlay in an image. The system **500** can be configured to perform and/or can perform one or more acts and/or functions, such as those described in this disclosure. The system **500** includes the feature map network **402** and the probability map network **404** described above with reference to FIG. 4.

As shown, the feature map network **402** receives an image **502** having an overlay **508**, which is one of a plurality of features in the image **502**. The image **502** may be described on a per-pixel basis using RGB values, for example. (While the accompanying figures are black-and-white drawings or grayscale images, the technology set forth herein is applicable to both color and grayscale images.) In general, the term “feature” may refer to any group of pixels in the image **502** having some common characteristic of interest, such as adjacent positioning with other pixels having a similar color, brightness, or other characteristic. A typical overlay is a feature having one or more shapes, logos, and/or characters (e.g. letters and/or numbers) that are co-located within a portion of the image **502** (generally not in the center of the image) and that may be, but need not be, of a single, homogeneous color, grayscale, and/or pattern, for example.

Upon receiving the image **502**, the feature map network **402** creates a feature map **504** having data characterizing one or more features in the image **502**, such as relative to other features in the image **502**. The feature map **504** may be a grid of feature vectors each having a corresponding length, such as 4096, for example.

The probability map network **404** receives the feature map **504** as an input and creates a probability map **506** that assigns a probability to the one or more features in the image **502**. The probability may specify a likelihood that a particular feature is an overlay (i.e. that a pixel or group of pixels (e.g. a patch) is part of an overlay). The probability map network **404** and creation of the probability map **504** is described below, with reference to FIG. 9. In the example of FIG. 5, the probability map **506** shows probability ranging

from black (zero or near-zero probability) to white (highly probable), such that a feature **510** (corresponding to a probable overlay location) is assigned a high probability (white or 1) of being an overlay (corresponding to the overlay **508** in the image **502**). One or more additional acts may be performed on features identified as having a high probability of being an overlay. FIGS. **6** and **7** describe examples of these additional acts that may be performed.

FIG. **6** is a simplified block diagram of a computing device **600** configured to detect overlays and perform one or more acts on the detected overlays. The computing device **600** includes the components of the computing device **400** illustrated in FIG. **4**, including the feature map network **402** and the probability map network **404**, and also includes two additional components: a masking component **602** and an inpainting component **604**.

With reference to FIG. **7**, the masking component **602** filters the probability map **506** to create an overlay mask **702**. In the case of images captured from a TV show, image components falling in the middle of the image are filtered out, since overlays do not typically appear in such a location (i.e. a target viewing portion of the image) of a TV show. Filtering out unlikely overlay locations can result in a smaller detection error rate, and can make subsequent review or processing easier. The resultant overlay mask **702** may be a binary mask, with only a small portion **704** (corresponding to the overlay **508** in the image **502**) indicating a probable location of a detected overlay.

The inpainting component **604** applies a conventional inpainting algorithm to create a modified inpainted image **706** with the overlay removed (i.e. replaced by an inpainted region **708**). For example, structural and/or textural inpainting algorithms can be used. Alternatively, the overlay can be replaced with pixels copied from an adjacent portion of the image **706** in a cloning or healing algorithm.

As an alternative to or in addition to inpainting, other acts may be performed on the detected overlay. For example, a detected overlay could be used to track compliance with advertising requirements. A frame counter or time counter could be incremented for images in which a compliant overlay is detected. As another example, some overlays may serve as hyperlinks in some images, such as in videos on media streaming sites, (e.g. clicking a hyperlink displayed while the video is playing will connect the user to a linked site or page, such as a particular channel on the media streaming site.). As yet another example, a detected overlay could be processed using OCR, which could help determine whether the detected overlay includes information that identifies a particular channel, show, or other attribute.

C. Feature Map Network

FIG. **8** is a simplified block diagram of an example feature map network **800**. In this example, an image **502** (e.g. described on a per-pixel basis using RGB values) input to the feature map network **800** to produce a feature map **504**. In the example of FIG. **8**, the feature map network **800** is a convolutional network.

The feature map network **800** includes a plurality of layers selected from convolutional layers and max-pooling layers. In general, each convolutional layer is a rectangular grid of neurons, each of which takes inputs from a weighted rectangular section of the previous layer. The max-pooling layers subsample small rectangular blocks from the previous convolutional layer to produce a single output from that block, corresponding to the maximum of the block they are pooling. The specific arrangement of convolutional layers and max-pooling layers for the feature map network **800** in the example of FIG. **8** will now be described.

As shown, the image **502** is received as an input (described as RGB values) to two $3 \times 3 \times 64$ convolutional layers **802**, the output of which is subsampled by a max pool layer **804**. The output of the max pool layer **804** is an input to two $3 \times 3 \times 128$ convolutional layers **806**, the output of which is subsampled by a max pool layer **808**. The output of the max pool layer **808** is an input to two $3 \times 3 \times 256$ convolutional layers **810**, the output of which is subsampled by a max pool layer **812**. The output of the max pool layer **812** is an input to two $3 \times 3 \times 512$ convolutional layers **814**, the output of which is subsampled by a max pool layer **816**. Two $1 \times 1 \times 4096$ convolutional layers **818** are applied to the output of the max pool layer **816** to create the feature map **504**, which serves as an output of the feature map network **800**. The feature map network **800** is shown in tabular form in Table 1, below.

The feature map network **800** is based on the Visual Geometry Group's (VGG's) ImageNet ILSVRC-2014 submission, Simonyan, Karen and Zisserman, Andrew, "Very Deep Convolutional Networks for Large-Scale Image Recognition," ICLR 2015 Conference Paper, Apr. 10, 2015, accessible as of at least July, 2017 at <https://arxiv.org/pdf/1409.1556.pdf>, the entirety of which is incorporated by reference herein. Several modifications were made to the 16-layer (VGG-16) model set forth in the aforementioned VGG paper; namely, the last max pool layer, the last three convolutional layers, and the last fully connected layer were removed, resulting in the 12-layer configuration shown in FIG. **8** and in Table 1, below. To generate the feature map **504**, consisting of a grid of feature vectors with length 4096, the remaining two fully connected layers are converted to 1×1 convolutional layers, as described in Sermanet et al., "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks," Feb. 24, 2014, accessible as of at least July, 2017, at <https://arxiv.org/pdf/1312.6229.pdf>, the entirety of which is incorporated by reference herein. This results in the feature map network configuration shown in FIG. **800** and in Table 1, below. Given that the feature map network **800** has four pooling (i.e. downsampling) layers, the feature map will be reduced in size compared to the original image and each feature vector in the grid will represent an image patch.

TABLE 1

Feature Map Network Configuration	
Input (RGB image)	
Conv	$3 \times 3 \times 64$
Conv	$3 \times 3 \times 64$
Max pool	2×2
Conv	$3 \times 3 \times 128$
Conv	$3 \times 3 \times 128$
Max pool	2×2
Conv	$3 \times 3 \times 256$
Conv	$3 \times 3 \times 256$
Conv	$3 \times 3 \times 256$
Max pool	2×2
Conv	$3 \times 3 \times 512$
Conv	$3 \times 3 \times 512$
Conv	$3 \times 3 \times 512$
Max pool	2×2
Conv	$1 \times 1 \times 4096$
Conv	$1 \times 1 \times 4096$

To train the feature map network **800**, a training set for the feature map network **800** may be created. Such a training set may include a set of images with various TV logo overlays superimposed thereon. For example, the Flickr 1M dataset, accessible at <http://press.liacs.nl/mirflickr/> as of at least July,

2017, can be used as a base and, for example, may be split into three parts for three different purposes: training (70%), validation (15%), and testing (15%). The overlays for the training set may be created by introducing (e.g. electronically pasting) semi-transparent or opaque images, such as logos, onto the images in the training set. For example, a set of 256 different channel logos can be pasted onto the images in the Flickr 1111 dataset described above. To provide diversity in the training set, the channel logos can be varied by grayscale or full color, opacity ($\alpha=\{25, 0.5, 0.75, 1\}$), size, and position in the image, for example. Creating a training set of images in this way allows relatively easy generation of a corresponding set of groundtruth masks. The groundtruth masks are the same size as the images, with all pixels equal to 0 where the overlaid logo does not overlap and equal to 1 where the overlaid logo does overlap. In one example, the training set images are resized so that the minimum dimension is 540 pixels and are then cropped to 256x256 pixels.

While FIG. 8 and Table 1 describe on feature map network based on VGG, other neural networks may alternatively be used, such as one based on GoogLeNet, a 22-layer deep network, AlexNet, ZF Net, ResNet, and other known neural networks.

D. Probability Map Network

FIG. 9 is a simplified block diagram of an example probability map network 900. The feature map 504 (from the feature map network 800) is an input to the probability map network 900, which produces the probability map 506, such as in the form of a modified image including a probable overlay location (e.g. feature 510 in FIG. 5). In the example of FIG. 9, the probability map network 900 uses two recurrent layers, one vertical and one horizontal, and combines (multiplies) recurrent layers to output a probability value between 0 and 1. The probability map 506 is used to decide where the overlay is, based on a threshold probability map (e.g. greater than 0.7 probability value corresponds to a probable overlay location).

The probability map network 900 uses available information to determine whether a particular pixel is part of an overlay. For example, if a pixel is highly probable to be part of an overlay, then adjacent pixels should also be likely to be part of the overlay. Accordingly, the probability map network is preferably a recurrent neural network—using long short term memory (LSTM) units, in particular. The feature map network 800 is able to produce deep convolutional representations of the images, while LSTM allows the probability map network 900 to keep track of and use long-term dependencies of these features. This tracking and use of long-term dependencies is described in Hochreiter et al., “Long Short-Term Memory,” *Neural Computation* 9(8): 1735-1780, 1997, accessible as of at least November, 2016, at http://deeplearning.cs.cmu.edu/pdfs/Hochreiter97_lstm.pdf, the entirety of which is incorporated by reference herein.

As illustrated in FIG. 9, the probability map network 900 utilizes bi-directional LSTMs 902 and 904, with peephole connections as described in Gers et al., “Recurrent Nets that Time and Count,” accessible as of at least July, 2017, at <ftp://ftp.idsia.ch/pub/juergen/TimeCount-IJCNN2000.pdf>, the entirety of which is incorporated by reference herein. However, unlike in the speech-recognition system described in Gers et al., the sequences in the probability map network 900 track dependencies vertically (LSTM 902) and horizontally (LSTM 904) in space, rather than forward and backward in time. The sequences for the vertical LSTM 902 are the feature vectors from the feature map 504 that run along

each column. The sequences for the horizontal LSTM 904 are the feature vectors from the feature map 504 that run along each row. The results from each sequence of the LSTMs 902 and 904 are passed through separate fully connected layers 906 and 908 with sigmoid nonlinearities 910 and 912. The resulting maps are multiplied together, as shown in block 912; meaning that a particular pixel must have both a high probability of being part of an overlay in the column-wise LSTM 902 and row-wise LSTM 904 to have a high probability overall. In the illustrated example, each LSTM 902 and 904 uses 250 cell states and gradient clipping is set to 0.1. In an alternative embodiment, four LSTMs could be used to improve the probability map 506. The illustrated (two LSTM) model only takes into account pixels that are positioned before the current pixel. Using four LSTMs would take into account the decisions for all four pixels surrounding the current pixel. Other configurations may be possible as well.

The example probability map network 900 described above is not the only possible configuration for detecting overlays in an image. The end application may dictate some configuration details, for example. In arriving at the example configuration of FIG. 9, a different configuration, referred to herein as a “Type I” configuration, was first tried. The Type I configuration determined patch-wise probability, i.e. the probability for a patch in the image to contain a logo. In this Type I network configuration, the final convolutional layer in the feature map has a filter size of 3x3, which results in a patch size of 48x48 with a 16 pixel stride. This also sets the number of outputs in the fully connected layers after the LSTMs to be 1 (i.e. a single probability for each patch). The Type I network therefore produces a low resolution probability map, which needs to be upsampled to give pixel-wise probability for the entire image.

To generate a full resolution probability map, a second configuration (a “Type II” configuration) was utilized, in order to obtain a probability vector for each patch. The Type II configuration uses an upscaling technique similar to what is described in Huval et al., “An Empirical Evaluation of Deep Learning on Highway Driving,” Apr. 17, 2015, accessible as of at least July, 2017, at <https://arxiv.org/pdf/1504.01716v3.pdf>, the entirety of which is incorporated by reference herein. This upscaling technique increases detection resolution to cover the entire input image size. The Type II configuration differs from the Type I configuration in several ways. One difference is the creation of smaller patches by reducing the final convolutional layer in the feature map network to 1x1 filters, as shown in Table 1 and described above with reference to FIG. 8. Using 1x1 filters provides a new patch size of 16x16 with a 16 pixel stride. The probability vector is a flattened version of the probabilities in each 16x16 patch. To obtain the entire probability map, the probability vectors are raveled into their respective patch positions as shown in FIG. 10, which illustrates 3x3 patches. As illustrated, the RUB probability vector 1002 is raveled to create a 3x3 RUB patch 1004. For this Type II configuration, set forth in FIG. 9 and shown in tabular form in Table 2 below, the fully connected layers following the LSTMs have 16x16=256 outputs. The Type II network provides a tighter detection of overlays and detect more overlays in general.

US 10,970,588 B2

11

TABLE 2

Type II Probability Map Network Feature Map	
LSTM 250 (across rows)	LSTM 250 (across columns)
Fully Connected 256 Sigmoid	Fully Connected 256 Sigmoid Multiply

The probability map network **900** may utilize a cost function, such as a mean squared error cost function, to arrive at an appropriate probability map **504**. In the Type I network described above, a probability is given for each patch. A mean squared error cost function is less strict than one utilizing binary cross entropy; therefore, the mean squared error cost function is more appropriate for the Type I network, as some patches will contain more of an overlay and some patches will contain less. The resulting probability should reflect how much of an overlay was present in a patch, rather than a hard decision about whether a patch contains any part of an overlay at all. For instance, a patch with only 1% of its pixels as part of an overlay should have a much lower probability than a patch that has 100% of its pixels as part of an overlay. Since there is a class imbalance of pixels (i.e. there are more pixels that do not belong to an overlay than ones that do), the cost function weights the error on the pixels that belong to the overlay class more heavily. A mean squared error cost function may be described as follows:

$$SE = (m - M)^2 \quad (1)$$

$$C = \frac{1}{N} (SE + \alpha (M \circ SE)) \quad (2)$$

Where SE is the squared error, m is the prediction, M is the groundtruth mask, N is the number of pixels in the image, α is the weight for the positive class, and \circ is the Hadamard Product (i.e. element-wise multiplication). Experimental results, set forth below, are based on $\alpha=2.5$ and L2 normalization with regularization weight of 0.0001.

For the Type I network, the weights of the feature map network were initialized with the weights that were used to achieve the results set forth in Simonyan et al., "Very Deep Convolutional Networks for Large-Scale Image Recognition," ICLR 2015 Conference Paper, Apr. 10, 2015. The trained weights from the Type I network's feature map were then used to initialize the feature map network for the Type II network. Without initializing the weights of the Type ii network with the Type I network weights, no training was observed.

III. Example Methods

FIG. **11** is a flow chart illustrating an example method **1100**.

At block **1102**, the method **1100** can include a processor applying a feature map network to an image to create a feature map, where the feature map comprises a grid of vectors characterizing at least one feature in the image. For example, applying the feature map network to the image may include applying a plurality of convolutional layers and max-pooling layers to the image, such as is shown and described with reference to FIG. **8**.

12

At block **1104**, the method **1100** can include the processor applying a probability map network to the feature map to create a probability map assigning a probability to the at least one feature in the image. The assigned probability corresponds to a likelihood that the at least one feature is an overlay. For example, the probability map network may be configured as shown and described with reference to FIG. **9**.

At block **1106**, the method **1100** can include the processor determining that the probability exceeds a threshold.

At block **1108**, the method **1100** can include, responsive to determining that the probability exceeds the threshold, performing a processing action associated with the at least one feature. For example, the processing action can be selected from the group consisting of removing the at least one feature from the image, inpainting the at least one feature, determining compliance with an advertising requirement, performing optical character recognition (OCR) on the at least one feature, determining a channel associated with the at least one feature, determining a show associated with the at least one feature, determining a hyperlink associated with the at least one feature, or accessing a hyper-linked object associated with the at least one feature.

The method **1100** can further include training the feature map network prior to the processor applying the feature map network to the image, such as by generating a plurality of groundtruth masks from a corresponding plurality of training images, where each of the plurality of training images has a respective overlay. According to one example, the plurality of training images can be created by placing a plurality of overlay images onto an original image set, where the overlay images are semi-transparent or opaque. In addition, the plurality of overlay images may be varied in at least one of grayscale intensity, color hue, color value, color intensity, opacity, size, or position in the original image set.

IV. Experimental Results

Two different test sets were used to verify the results. One test set is the artificial test set, which is created in a similar way to the training test set with the remaining 15% of the Flickr 1M dataset that were set aside. The other test set is images taken from actual TV shows that have aired. The artificial test sets allows for calculating the pixel-wise accuracy of the predictions, while the test set from actual TV shows illustrates viability in real world scenarios.

A. Artificial Test Set

The artificial test set uses the 15% of the Flickr 1M dataset (i.e. 149999 images) and is created similarly to the training set, described above. The same 256 different channel logos that were used in the training set are pasted onto the images, varying by grayscale or full color, opacity ($\alpha=\{0.25, 0.5, 0.75, 1\}$), size, and position in the image, for example. The main difference is that after the minimum dimension of the image is resized to 540 pixels, the image is cropped to 288x288 instead of 256x256.

As the Type II network configuration builds on top of Type I (see the description above with reference to FIG. **9**), the following results focus on the Type II network. To create masks to compare the prediction with the groundtruth, simple thresholding is performed on the probability map from the network. Pixels whose probability fall above this threshold (e.g. 0.7) are considered to be part of an overlay and pixels that do not are part of the background. Mean average precision (mAP) and area under the ROC curve (AuC) are suitable evaluation metrics, although others are possible as well.

US 10,970,588 B2

13

B. Real World Test Set

A set of 51 images were captured from actual TV shows playing on TV. Using this set shows that this model can be used beyond the artificial data set and in real world applications. The images have a variety of different overlays ranging from logos to text to show ratings. In the 51 captured images, there were 58 total annotated overlays. These included 48 images with at least one or more overlays and three images with no overlays at all. These images are resized so that their minimum dimensions are 540 pixels, and no cropping is performed. However, since these are stills captured from TV, groundtruth cannot be created in the same way as for the artificial set.

In the real world set, a two-step thresholding scheme is performed on the probability map, similar to the one for canny edge detection. There is a high threshold which is first used. Then a lower threshold is used, but only components that touch components which are above the high threshold are kept. The high threshold used is 0.9, and low threshold is 0.25. This two-step thresholding allows the mask to close any gaps between high probability areas. Furthermore, for this case only, and as illustrated in FIG. 9, any components found in the middle of the image were removed, considering that broadcast stations are unlikely to place any overlays in the middle of their content.

Since there are no groundtruth masks available for these images in the same way as for the artificial test images, bounding boxes were annotated around each overlay in the images. Text and logos that are meant to be read together are grouped together into one bounded box. These images are analyzed by determining whether the bounded box of the detection region and groundtruth bounded boxes have an intersection over union of greater (IOU) than 0.5.

To show the improvements of the Type II network over the Type I network, we used the real world test set, as this test set is tougher than the artificial test set. Table 3, below, shows that the Type II network works better in this case. Since the Type II probability network is able to account for all pixels of the input image, its detection results are more accurate. Even though the bounding boxes generated from both network configurations may fit the intersection over union threshold of 0.5, the detections from the Type II configuration are closer to the actual overlay.

TABLE 3

Comparison of Type I and Type II network on the real world test set		
	Type I	Type II
True Positives	26	48
False Positives	23	10
False Negatives	32	10
Precision	0.5306	0.8276
Recall	0.4482	0.8276
F score	0.4859	0.8276

IV. Example Variations

Although some of the acts and/or functions set forth in this disclosure have been described as being performed by a particular entity, the acts and/or functions can be performed by any entity, such as those entities described in this disclosure. Further, although the acts and/or functions have been recited in a particular order, the acts and/or functions need not be performed in the order recited. However, in some instances, it can be desired to perform the acts and/or

14

functions in the order recited. Further, each of the acts and/or functions can be performed responsive to one or more of the other acts and/or functions. Also, not all of the acts and/or functions need to be performed to achieve one or more of the benefits provided by this disclosure, and therefore not all of the acts and/or functions are required.

Although certain variations have been discussed in connection with one or more examples of this disclosure, these variations can also be applied to all of the other examples of this disclosure as well.

Although select examples of this disclosure have been described, alterations and permutations of these examples will be apparent to those of ordinary skill in the art. Other changes, substitutions, and/or alterations are also possible without departing from the invention in its broader aspects as set forth in the following claims.

The invention claimed is:

1. A method comprising:

a processor applying a feature map network to an image to create a feature map, wherein the feature map comprises a grid of vectors characterizing at least one feature in the image;

the processor applying a probability map network to the feature map to create a probability map assigning a probability to the at least one feature in the image, wherein the assigned probability corresponds to a likelihood that the at least one feature is an overlay;

the processor determining that the assigned probability exceeds a threshold probability value; and

responsive to the processor determining that the assigned probability exceeds the threshold probability value, the processor (i) determining compliance with an advertising requirement, (ii) performing optical character recognition (OCR) on the at least one feature, (iii) determining a channel associated with the at least one feature, (iv) determining a show associated with the at least one feature, (v) determining a hyperlink associated with the at least one feature, or (vi) accessing a hyperlinked object associated with the at least one feature.

2. The method of claim 1, further comprising:

responsive to the processor determining that the assigned probability exceeds the threshold probability value, the processor (i) removing the at least one feature from the image or (ii) inpainting the at least one feature.

3. The method of claim 1, further comprising:

training the feature map network prior to the processor applying the feature map network to the image.

4. The method of claim 3, wherein training the feature map network comprises generating a plurality of groundtruth masks from a corresponding plurality of training images, wherein each training image of the plurality of training images has a respective overlay.

5. The method of claim 4, further comprising:

creating the plurality of training images by placing a plurality of overlay images onto an original image set, wherein each overlay image of the plurality of overlay images is semi-transparent or opaque.

6. The method of claim 5, further comprising:

varying the plurality of overlay images by at least one of grayscale intensity, color hue, color value, color intensity, opacity, size, or position in the original image set.

7. The method of claim 1, wherein applying the feature map network to the image comprises applying a plurality of convolutional layers and max-pooling layers to the image.

8. The method of claim 7, wherein the image is described on a per-pixel basis using RGB values, and wherein the

US 10,970,588 B2

15

plurality of convolutional layers and max-pooling layers comprises the following, in order:

- two convolutional 3×3 64-length layers;
- a first max-pooling 2×2 layer;
- two convolution 3×3 128-length layers;
- a second max-pooling 2×2 layer;
- three convolution 3×3 256-length layers;
- a third max-pooling 2×2 layer;
- three convolution 3×3 512-length layers;
- a fourth max-pooling 2×2 layer; and
- two convolution 1×1 4096-length layers.

9. The method of claim 1, wherein applying the probability map network to the feature map comprises applying the following, in order:

- a first LSTM across rows and a second LSTM across columns;
- two fully connected layers;
- two sigmoid layers; and
- a multiplication layer.

10. A non-transitory computer-readable medium having stored thereon program instructions that, upon execution by a processor, cause the processor to perform a set of acts comprising:

applying a feature map network to an image to create a feature map, wherein the feature map includes data characterizing a feature in the image relative to other features in the image;

applying a probability map network to the feature map to create a probability map assigning a probability to the feature in the image, wherein the assigned probability corresponds to a likelihood that the feature is an overlay;

determining that the assigned probability exceeds a threshold probability value; and

responsive to determining that the assigned probability exceeds the threshold probability value, (i) determining compliance with an advertising requirement, (ii) performing optical character recognition (OCR) on the at least one feature, (iii) determining a channel associated with the at least one feature, (iv) determining a show associated with the at least one feature, (v) determining a hyperlink associated with the at least one feature, or (vi) accessing a hyperlinked object associated with the at least one feature.

11. The non-transitory computer-readable medium of claim 10, the set of acts further comprising:

responsive to determining that the assigned probability exceeds the threshold probability value, (i) removing the at least one feature from the image or (ii) inpainting the at least one feature.

12. The non-transitory computer-readable medium of claim 10, the set of acts further comprising:

training the feature map network prior to the processor applying the feature map network to the image, wherein training the feature map network comprises generating a plurality of groundtruth masks from a corresponding plurality of training images, wherein each training image of the plurality of training images has a respective overlay.

13. The non-transitory computer-readable medium of claim 12, wherein each groundtruth mask of the plurality of groundtruth masks is the same size as its corresponding training image, the set of acts further comprising:

setting each pixel in a respective groundtruth mask of the plurality of groundtruth masks to 1 if the pixel is overlapped by the respective overlay or to 0 if the pixel is not overlapped by the respective overlay.

16

14. The non-transitory computer-readable medium of claim 12, the set of acts further comprising:

creating the plurality of training images by placing a plurality of overlay images onto an original image set, wherein each overlay image of the plurality of overlay images is semi-transparent or opaque.

15. The non-transitory computer-readable medium of claim 10, wherein applying the feature map network to the image comprises applying a plurality of convolutional layers and max-pooling layers to the image.

16. The non-transitory computer-readable medium of claim 15, wherein the image is described on a per-pixel basis using RGB values, and wherein the plurality of convolutional layers and max-pooling layers comprises the following, in order:

- two convolutional 3×3 64-length layers;
- a first max-pooling 2×2 layer;
- two convolution 3×3 128-length layers;
- a second max-pooling 2×2 layer;
- three convolution 3×3 256-length layers;
- a third max-pooling 2×2 layer;
- three convolution 3×3 512-length layers;
- a fourth max-pooling 2×2 layer; and
- two convolution 1×1 4096-length layers.

17. The non-transitory computer-readable medium of claim 10, wherein applying the probability map network to the feature map comprises applying bi-directional LSTMs with peephole connections to the feature map.

18. The non-transitory computer-readable medium of claim 17, wherein the bi-directional LSTMs with peephole connections comprises the following, in order:

- a first LSTM across rows and a second LSTM across columns;
- two fully connected layers;
- two sigmoid layers; and
- a multiplication layer.

19. A system, comprising:

a processor; and

a non-transitory computer-readable medium having stored thereon program instructions that, upon execution by the processor, cause the processor to perform a set of acts comprising:

training a feature map network by generating a plurality of groundtruth masks from a corresponding plurality of training images, wherein each of the plurality of training images has an overlay with at least one of a respective grayscale intensity, color hue, color value, color intensity, opacity, size, or position that differs relative to other overlays in the plurality of training images;

applying the trained feature map network to an image to create a feature map, wherein the feature map includes data characterizing a feature in the image relative to other features in the image, wherein the feature map network comprises a very deep convolutional neural network having a plurality of convolutional layers and max-pooling layers;

applying a probability map network to the feature map to create a probability map, wherein the probability map network comprises at least two recurrent layers including at least one vertical recurrent layer and at least one horizontal recurrent layer that are multiplied to provide an output indicative of a probability that the feature in the image is an overlay;

determining that the probability exceeds a threshold probability value; and

US 10,970,588 B2

17

responsive to determining that the probability exceeds
the threshold probability value, (i) determining com-
pliance with an advertising requirement, (ii) per-
forming optical character recognition (OCR) on the
feature, (iii) determining a channel associated with
the feature, (iv) determining a show associated with
the feature, (v) determining a hyperlink associated
with the feature, or (vi) accessing a hyperlinked
object associated with the feature.

20. The system of claim 19, the set of acts further
comprising:

responsive to determining that the probability exceeds the
threshold probability value, (i) removing the feature
from the image or (ii) inpainting the feature.

* * * * *

15

18

EXHIBIT B

**IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE**

THE NIELSEN COMPANY (US), LLC,)	
)	
<i>Plaintiff,</i>)	
v.)	C.A. No. _____
)	
HYPHAMETRICS, INC.,)	
)	
<i>Defendant.</i>)	
)	

Declaration of Pierre Moulin

I, Pierre Moulin, declare as follows:

1. I have been retained by The Nielsen Company (US), LLC ("Nielsen") to provide an independent opinion regarding U.S. Patent No. 10,970,588 ("the '588 Patent").

2. I am not an employee of Nielsen or any affiliate or subsidiary of Nielsen. Nor do I have a financial interest in Nielsen or the outcome of this case. I am being compensated for my work at my standard consulting rate. My compensation is not dependent on the outcome of this case or on the content of my opinions.

3. The '588 Patent relates to detection of overlays in images. In my opinion, a person of ordinary skill in the art in the field of the '588 Patent would have a working knowledge of basic computer technology and video/image processing technology. The person would gain this knowledge through either (i) an undergraduate Bachelor of Science degree in Computer Science or Electrical Engineering or a comparable field, and at least three years of work experience in relevant fields; or (ii) a Master's degree in Computer Science or Electrical Engineering and at least one year of work experience in relevant fields. I base these statements on my experience, including my knowledge of colleagues and others at the time of the invention of the '588 Patent.

4. Unless otherwise noted, the statements made in this Declaration are based on my personal knowledge, and if called to testify about this declaration, I could and would do so competently and truthfully.

5. A detailed record of my professional qualifications is attached to this Declaration as Exhibit A.

6. I have reviewed the '588 Patent and the materials that are listed in Exhibit B to this Declaration. My opinions are, in part, a result of my review of those materials. My opinions are also a result of my years of experience in the field. Moreover, my experience has informed my choices of materials to review.

7. I am not a legal expert and offer no opinions on the law.

8. I am a Professor at the University of Illinois at Urbana-Champaign ("UIUC") in the Department of Electrical and Computer Engineering, a Research Professor in the Coordinated Science Laboratory, and a faculty member in the Beckman Institute for Advanced Science and Technology and its Image Formation and Processing Group. I am also a member of the Information Trust Institute and the founding director of the Center for Information Forensics, a multidisciplinary research center.

9. I am an experienced researcher and educator in the field of electrical and computer engineering, with expertise in signal processing, audio and video/image processing, information theory, machine learning, computer vision, and large data systems.

10. I have experience and expertise in numerous areas including information theory, audio and video/image processing, computer vision, statistical signal processing and modeling, decision theory, information hiding and authentication, and the application of multi-resolution signal analysis, optimization theory, and fast algorithms to these areas. As part of my professional

research, I have studied machine learning as it applies to both video/image and audio electronic content.

11. I have more than 35 years of experience in the field of electrical and computer engineering since receiving my bachelor's degree in 1984. I have 27 years of experience as a professor in the Electrical and Computer Engineering Department of UIUC.

12. During this time, I have led, overseen, and contributed to numerous research projects involving technical concepts that are closely related to the technology at issue in the '588 Patent.

13. I have taught numerous courses on technology relating to audio and video/image processing, digital signal processing, information theory, and detection and estimation theory. I have developed courses in image and video processing, machine learning, and large data computing.

14. In my career field, I have received numerous accolades for my contributions to electrical and computer engineering, including being elected as Fellow of the IEEE in 2003. In 2018, I was honored to receive the Ronald W. Pratt Faculty Outstanding Teaching Award. I served on the IEEE Signal Processing Society Board of Governors from 2005-2007 and the IEEE Information Theory Society Board of Governors from 2016-2018. I also served as IEEE Signal Processing Society Distinguished Lecturer from 2012-2013.

15. I received my D.Sc. in Electrical Engineering (Washington University, St. Louis, 1990); M.Sc. in Electrical Engineering (Washington University, St. Louis, 1986); and Ingénieur Civil Electricien (Faculté Polytechnique de Mons, Belgium, 1984). The D.Sc. is equivalent to a Ph.D. and is recognized as such within academic and research communities.

16. After receiving my D.Sc. degree, I worked as a Research Scientist at Bell Communications Research, Morristown, NJ from 1990-1995.

17. My work as an academic began in 1996, when I joined the University of Illinois as Assistant Professor. In 1999, I was promoted to Associate Professor, and, in 2003, I was promoted to the position of Professor in the department of Electrical and Computer Engineering and affiliate Professor in the department of Statistics.

18. During sabbaticals from University of Illinois, I have held visiting positions at the National Taiwan University (2019), the Chinese University of Hong Kong (2010 and 2014), MIT (2005), and Microsoft Research (2001).

19. I have received funding from numerous agencies, foundations, and companies for my research on electrical and computer engineering, including image and signal processing. The sources of funding for this research include the National Science Foundation (NSF), the Defense Advanced Research Projects Agency (DARPA), Microsoft, HP Labs, Xerox, and the North Atlantic Treaty Organization (NATO).

20. I have served in a number of leadership roles for the IEEE, including as an Editorial Board member for the *Proceedings of the IEEE* (2007-2012), as Co-founder and Editor-in-chief for *IEEE Transactions on Information Forensics and Security* (2005-2008), and as Area Editor for *IEEE Transactions on Image Processing* (2002-2006).

21. I have been an invited lecturer at academic institutions and organizations including MIT, Stanford, Berkeley, Carnegie-Mellon, Harvard, Bell Labs, IBM Research, and Microsoft Research, among others.

22. I have graduated thirteen Ph.D. students and educated 9 post-doctoral researchers. Three of my Ph. D. students became associate professors upon graduation, and one became a

professor upon graduation. My other Ph.D. students have joined companies such as Intel Research, Qualcomm, and Google. Seven of my post-doctoral researchers began careers in academia in professorial positions.

23. I have researched and written about video and audio processing, video and audio hashing, video and audio identification, large data systems, and other topics related to video and audio processing.

24. During the course of my career, I have authored or co-authored more than 65 scientific articles, one book, and six book chapters. I have presented and/or published 194 conference papers. I have also been awarded two U.S. patents. These, and other aspects of my qualifications, are summarized in my curriculum vitae accompanying this declaration as Exhibit A.

25. The '588 Patent relates to, among other things, the detection of semitransparent or opaque overlays (such as logos or text) in images (such as stills captured from television programs) and taking action based on that detection. (*See* '588 Patent, 2:48-51.) In certain aspects of the '588 Patent, a probability map is generated based on an image, and an overlay is identified. (*See* '588 Patent, 2:58-63.) Based on the identification of the overlay, actions such as channel recognition may be taken. (*See* '588 Patent, 2:63-67.)

26. As the specification of the '588 Patent points out, "[a] typical overlay is a feature having one or more shapes, logos, and/or characters (*e.g.* letters and/or numbers) that are co-located within a portion of the image (generally not in the center of the image) and that may be, but need not be, of a single, homogeneous color, grayscale, and/or pattern, for example." ('588 Patent, 6:47-52.)

27. The invention of the '588 Patent "provides an improvement to the technical field of image processing, and, in particular, to improving speed and reliability of detecting overlays in images, with reduced or eliminated human oversight." ('588 Patent, 3:18-22.)

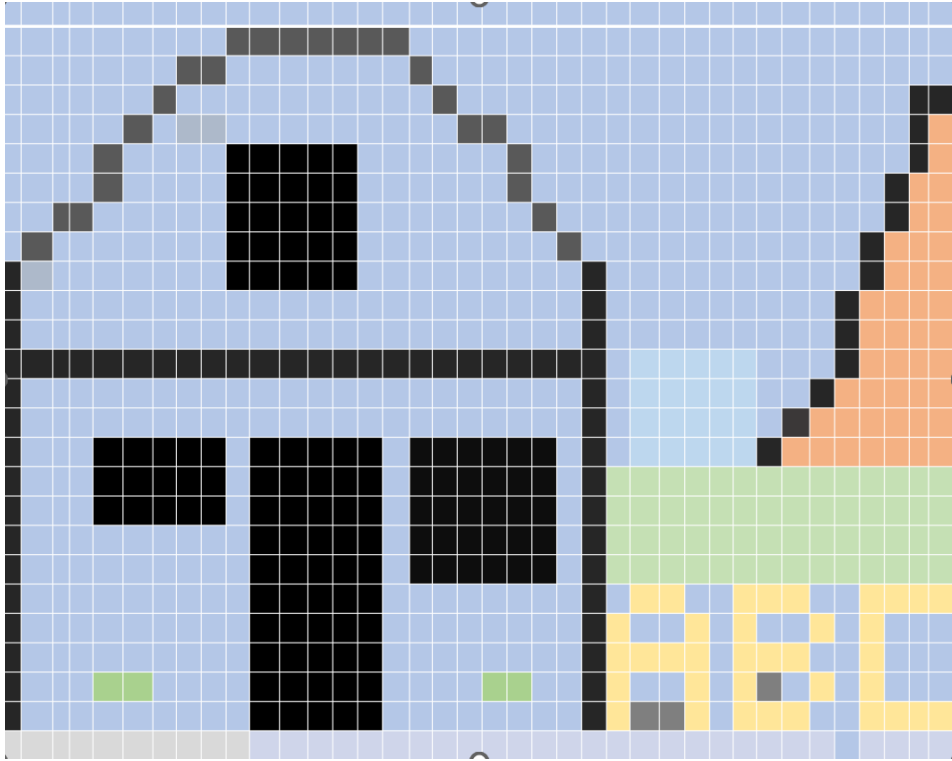
28. Claim 1 of the '588 Patent recites a specific method of overlay detection and subsequent action. Claim 10 of the '588 Patent recites a specific non-transitory computer-readable medium having program instructions for a particular approach to overlay detection and subsequent action.

29. Claim 1 recites applying a feature map network to an image to create a feature map, wherein the feature map comprises a grid of vectors characterizing at least one feature in the image (the "Feature Map Element"). The specification of the '588 Patent describes examples of how this may be accomplished at 7:50-67, 8:1-67, and 9:1-24. Claim 10 recites an element that is identical to the Feature Map Element for purposes of this declaration.

30. To visualize how a feature map works, a simple example is set forth below. The input to the feature map network is a photographic image of a house. The image includes an overlay of the letters "ABC" in the bottom right corner, which is somewhat hard to see owing to clutter and similar colors in that region of the original image.



31. The feature map computed by the example network is the 40 by 25 array of pixels shown below. Each of the pixels of the feature map represents salient features of the corresponding image region, such as average intensity (for a flat homogeneous region such as a house wall, or a relatively homogeneous region such as a door or a window), orientation of a strong edge (which could be horizontal, vertical, or diagonal), or a letter of the alphabet. For illustration purposes, different pixels of the feature map shown below may show different elements of the corresponding vectors (*e.g.*, a first pixel may show the first component of its vector, while a second pixel may show the tenth component of its vector), so that each pixel shows a notable feature of the corresponding region, thus facilitating an intuitive understanding of the relationship between the image above and the feature map below. Some color information is preserved (*e.g.*, the bushes appear green in the feature map). Color is also used in the example to enhance the visibility of some image elements. For instance, detected letters are rendered using a yellow color, and strong edges are rendered using a dark grey color. Finally, note that since the feature map is computed from the input image, it may contain noise due to clutter and shape and intensity variations within the image.



32. Claim 1 further recites creating a probability map for the image that assigns a probability to the at least one feature in the image, where the assigned probability corresponds to a likelihood that the at least one feature is an overlay (the "Probability Map Element"). ('588 Patent, Claim 1.) The specification of the '588 Patent describes examples of how this may be accomplished at 9:25-67, 10:1-67, and 11:1-54. Claim 10 recites an element that is identical to the Probability Map Element for purposes of this declaration.

33. Continuing the above example, a probability map (shown below) is computed from the feature map and expressed as a percentage, where 1% indicates very low probability that the portion of the image (*i.e.*, the small box) is part of an overlay, and 100% indicates absolute certainty (although absolute certainty generally cannot be achieved, due to inevitable noise in the feature map). Probabilities are calculated for groups of pixels, where each defined area (*i.e.*, small box) in the probability map represents a 5 x 5 block of pixels. In the example, the letter C in the

bottom right corner contains essentially no noise and is deemed to have the highest probability (99%) of being part of an overlay. In contrast, the letter A is somewhat obscured in the image and corrupted by slight noise (two noisy pixels) and is therefore deemed to have lower probability (82%) of being part of an overlay. The letter B is also somewhat corrupted, but not as much as the letter A, and thus, the probability that the letter B is part of an overlay is deemed to be 89% (*i.e.*, a probability that is between the probability for A and the probability for C). The other parts of the image featuring walls, doors, windows, roof, bushes, etc. are deemed to be highly unlikely to belong to a text overlay. While the associated probabilities vary across the image depending on the level of potentially confusing detail and noise, those probabilities are found to be below 10%.

1	3	1	3	1	1	1	1
4	1	5	1	3	1	1	3
2	8	2	6	2	1	1	3
3	2	7	2	2	1	3	3
2	4	7	3	3	82	89	99

34. Claim 1 and Claim 10 also recite determining that the assigned probability exceeds a threshold probability value (the "Probability Threshold Element"). The specification of the '588 Patent describes an example of this at 9:35-38.

35. Continuing our example, the probability threshold is set to a value of 75% to ascertain the presence of the overlay. Accordingly, the three boxes in the bottom right corner of the array are deemed to be part of a text overlay, and that the remaining thirty seven boxes are not.

36. Finally, Claim 1 and Claim 10 recite taking particular action responsive to determining that the assigned probability exceeds the probability threshold (the "Action Element"). Specifically, the Action Element recites (i) determining compliance with an advertising

requirement, (ii) performing optical character recognition (OCR) on the at least one feature, (iii) determining a channel associated with the at least one feature, (iv) determining a show associated with the at least one feature, (v) determining a hyperlink associated with the at least one feature, or (vi) accessing a hyperlinked object associated with the at least one feature.

37. I understand that the '588 Patent is based on two provisional patent applications filed in November 2016. As of that time (*i.e.*, in the prior art), the Probability Map Element and the Probability Threshold Element were not routine, conventional, or well-understood.

38. Even though the prior art describes substantial efforts in general image classification, the problem of classification with localization (*i.e.*, showing the location of an object of interest in an image using, for example, a bounding box), as is required for overlay detection, adds a level of complexity relative to the basic image classification problem. Accordingly, the problem of detecting overlays is very specialized and has received limited attention in the prior art.

39. In the prior art, probability maps (as recited in the Probability Map Element and the Probability Threshold Element) were not used to detect overlays. Also, in the prior art, only regions of interest of an image were analyzed. The use of probability maps and the analysis of an image (as recited in the Probability Map Element and the Probability Threshold Element) are significant technological improvements over the prior art. The Probability Map Element and the Probability Threshold Element were not well-understood, routine, or conventional in the prior art.

40. The prior art approach of examining only regions of interest is less accurate than examining the image (as recited in Claims 1 and 10 of the '588 Patent) because finding regions of interest is an error-prone process. Accordingly, the Probability Map Element and the Probability Threshold Element of Claim 1 are an improvement over the prior art. As stated above, the

Probability Map Element and the Probability Threshold Element were not well-understood, routine, or conventional in the prior art.

41. In addition, instead of assigning probabilities (and determining whether they exceed thresholds) as recited in the Probability Map Element and the Probability Threshold Element, in the prior art, binary linear classifiers were assigned to the above-described regions of interest. The output of such classifiers does not allow the ascertaining of actual probability. The use of probability maps and probability thresholds (as recited in the Probability Map Element and the Probability Threshold Element) inherently results in a more accurate detection of an overlay (*i.e.*, having lower error) than the prior art approach of using of binary linear classifiers. Thus, the Probability Map Element and the Probability Threshold Element are an improvement over the prior art. As stated above, the Probability Map Element and the Probability Threshold Element were not well-understood, routine, or conventional in the prior art.

42. The Action Element of Claim 1 of the '588 Patent was not routine, conventional, or well-understood in the prior art. In the prior art, the actions recited in the claim were not taken as a result of threshold probability of the presence of an overlay being exceeded or as a result of applying an analysis to an image.

43. I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct.

Executed on: February 3, 2023

Pierre Moulin

Pierre Moulin

EXHIBIT A

PIERRE MOULIN

University of Illinois
Coordinated Science Laboratory
1308 W. Main St.
Urbana, IL 61801

tel (217) 244-8366
fax (217) 244-8371
[Email: moulin@ifp.uiuc.edu](mailto:moulin@ifp.uiuc.edu)
WWW: www.ifp.uiuc.edu/~moulin

Education	D.Sc. in Electrical Engineering, <i>Washington University, St Louis</i>	08/1987–05/1990
	M.Sc. in Electrical Engineering, <i>Washington University, St Louis</i>	08/1985–05/1986
	Ingénieur Civil Electricien, <i>Faculté Polytechnique de Mons, Belgium</i>	09/1979–07/1984

Professional Experience

Professor of ECE and Statistics, <i>University of Illinois</i> , Urbana, IL	08/2003–present
Associate Professor, <i>University of Illinois</i> , Urbana, IL	08/1999–08/2003
Assistant Professor, <i>University of Illinois</i> , Urbana, IL	01/1996–08/1999
Research Scientist, <i>Bell Communications Research</i> , Morristown, NJ	10/1990–12/1995
Research Assistant, <i>Washington University</i> , St Louis, MO	10/1987–05/1990
Research Engineer, <i>Faculté Polytechnique de Mons, Belgium</i>	10/1984–06/1985
Summer Intern, <i>Standard Telecommunications Laboratories Ltd.</i> , U.K.	07/1983–08/1983

Professional Activities

Senior Editorial Board member of *IEEE Journal of Selected Topics in Signal Processing* 2019—present
 Editorial Board member for *Proceedings of IEEE*, 2007-2012
 Co-Founder and Editor in Chief for *IEEE Transactions on Information Forensics and Security*, 2005-2008
 Area Editor for *IEEE Transactions on Image Processing*, 2002-2006.
 Guest Editor for *IEEE Transactions on Signal Processing* supplements on secure media, 2004-2005.
 Guest Associate Editor for *IEEE Transactions on Signal Processing*'s special issue on Data Hiding, Apr. 2003.
 Guest Associate Editor for *IEEE Transactions on Information Theory*'s special issue on Information-theoretic imaging, Aug. 2000
 Associate Editor for *IEEE Transactions on Image Processing*, 1999—2002
 Associate Editor for *IEEE Transactions on Information Theory*, 1996—1998
 Member, IEEE Image and Multidim. Signal Processing Technical Committee, 1998-2003
 Co-chair, Allerton Conference 2007, 2008
 Member, Technical Program Committee of miscellaneous major IEEE conferences
 Invited organizer, NSF Workshop on Signal Authentication, Orlando, FL, 2002.
 Organizer, special sessions on watermarking at Allerton (10/01, 02, 03) and on information-theoretic imaging at Asilomar (11/02)
 Invited lectures at MIT, Stanford, Berkeley, Carnegie-Mellon, Harvard, Illinois, Purdue, Michigan, UC San Diego, UC Santa Barbara, Maryland, National U. of Singapore, Nanyang Tech. U. (Singapore), Chinese U. Hong Kong, Hong Kong U. of Science and Technology, Hong Kong Polytechnic, National Taiwan U., Academia Sinica, National Chiao Tung U. (Taiwan), KAIST (Korea), INRIA Sophia-Antipolis (France), Delft (Netherlands), Louvain (Belgium), Tech. U. Munich (Germany), Bell Labs, IBM Research, Microsoft Research, KLA-Tencor, Qualcomm.

Keynote/Plenary talks at Int. Workshop on Digital Watermarking (IWDW), Seoul, Korea, 2002; Int. Symp. on Image and Signal Processing and Analysis (ISPA), Rome, 2003; Wav-ila Challenge, Barcelona, Spain, 2005; IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Toulouse, France, 2006; Workshop on Multimedia Content Representation, Classification and Security (MRCS), Istanbul, Turkey, 2006; 3rd Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), Kaohsiung, Taiwan, 2007; IEEE Int. Conf. on Image Processing (ICIP), Brussels, Belgium, 2011; EUROCON, Zagreb, Croatia, 2013; ICSIPA, Melaka, Malaysia, 2013; VCIP, Singapore, 2015; ISPACS, Taipei, 2019.

Tutorial lecturer at ICIP'01, ICASSP'02, ICIP'04, and ISIT'06.

Co-Chair, IEEE Info Theory Workshop in Detection, Classif. & Imaging (Santa Fe, 2/99); Allerton Conf. (Monticello, IL, 9/06, 9/07); IEEE Workshop on Information Forensics and Security (Tenerife, Spain, 11/12); Tech. Program co-chair, IEEE Symp. on Information Theory (Hong Kong, 6/15).

Member, *IEEE Int. Conf. on Image Processing* Organizing Committee (Chicago 1998; Technical Program Co-chair, Brussels 2011)

Reviewer and panelist for National Science Foundation, reviewer for IEEE Trans. on SP, IP, IT, COM, NN and JOSA, CVGIP, JVCIP, Addison-Wesley, IEEE Press, SIAM, and Annals of Statistics.

Consultant and expert witness

Founding Director of *Center for Information Forensics*, UIUC.

Honors and Awards

Ronald W. Pratt Faculty Outstanding Teaching Award, 2018

IEEE Information Theory Society Board of Governors, 2016—2018

IEEE Signal Processing Society Distinguished Lecturer, 2012—2013

UIUC Sony Faculty Scholar, 2005—2007

IEEE Signal Processing Society Board of Governors, 2005—2007

IEEE Fellow, 2003

Beckman Associate in UIUC Center for Advanced Study, 2003

Co-author, IEEE Signal Processing Society 2002 Young Author Best Paper Award

IEEE Signal Processing Society 1997 Best Paper Award

NSF Career award, 1998-2001

UIUC Incomplete List of Teachers Rated as Excellent, 1996, 1999, 2000, 2005, 2007, 2009, 2011, 2012, 2016, 2019

Rotary International Scholar, 1985-1986

A. Dosin Award Recipient, 1984

Co-winner 1989 US Amateur Team Chess Championship

University Service

Chair, ECE Faculty Search Committee, 2019—Present

ECE Promotions and Tenure Committee, 2009-Present

Chair, ECE Signal Processing and Circuits Area, 2007-2017

ECE Faculty Search Committee, 2006—2012 and 2015—Present

ECE Graduate Committee, 1997—2008 and 2011—2014

ECE Representative to Research Board, 2007-2008

Elected Member, UIUC Faculty Senate, 2004-2005

Member, Information Trust Institute educational committee, 2004 - 2006

Elected Member, ECE Faculty Advisory Committee, 2002-2003

Member, Computer Resources and Education Committee, 1999—2002

Chair, ECE Graduate Seminar Committee, 1998—1999

Member, ECE Graduate Seminar Committee, 1996—2003

Ph.D. Graduates

	Name	Year	Placement
1.	Juan (Julia) Liu	2001	Research Scientist, PARC, Palo Alto, CA
2.	Prakash Ishwar	2002	Prof., Boston U.
3.	Kivanc Mihcak	2002	Assoc Prof., Bogazici U., Istanbul, Turkey
4.	Shawn C. Herman	2002	Numerica Inc., Fort Collins, CO.
5.	Tie Liu	2006	Assoc. Prof., Texas AM
6.	Ying (Grace) Wang	2006	Flarion Qualcomm, NJ
7.	Negar Kiyavash (co-advisor)	2006	Assoc. Prof., UIUC
8.	Maha El Choubassi	2008	Intel Research, Santa Clara, CA
9.	Shankar Sadasivam	2011	Qualcomm, San Diego, CA
10.	Yen-Wei Huang	2013	Google
11.	Honghai Yu	2015	A*STAR, Singapore
12.	Patrick Johnstone	2017	Postdoc, Rutgers U.
13.	Amish Goel	2022	

Postdocs

	Name	Year	Placement
1.	Aaron Lanterman	1998-2001	Prof, Georgia Tech
2.	Sviatoslav Voloshnykovskiy	1999	Assoc. Prof., U. of Geneva
3.	Jong C. Ye	1999-2001	Prof, KAIST, Korea
4.	Natalia Schmid	2001	Assoc. Prof., U. of West Virginia
5.	Bing Bing Ni	2010-2015	Asst. Prof., Shanghai Jiaotong U., China
6.	Jiwen Lu	2012-2015	Asst. Prof., Tsinghua U., Beijing, China
7.	Gang Wang	2012-2015	Assoc. Prof., Nanyang Tech. U., Singapore
8.	Zhang Le	2016-	ADSC Singapore
9.	Jagan Varadarajan	2013 - 2016	ADSC Singapore

Funding

NSF, DARPA, ARO, ONR, NATO, A*STAR, Microsoft, HP Labs, Xerox, KLA-Tencor

Sabbatical Leaves

National Taiwan University, 2019

Chinese University of Hong Kong, 2010 and 2014

MIT, 2005

Microsoft Research, 2001

Course Development

1. ECE418, Introduction to Image and Video Processing, 1996-1998
2. ECE544 Wavelets, 1998
3. ECE544 Statistical Image and Video Processing, 2006-2008
4. ECE544 Statistical Learning, 2012, 2015, 2016
5. ECE598PM (now ECE566) Computational Inference and Learning, 2014-2017
6. ECE398BD (now ECE314) Making Sense of Big Data, 2014.

Other Teaching

Introduction to Image and Video Processing (ECE418), Digital Signal Processing II (ECE551), Information Theory (ECE563), Random Processes (ECE534), Detection and Estimation Theory (ECE561)

Journal Papers

- A. Goel and P. Moulin, "Fast Locally Optimal Detection of Targeted Universal Adversarial Perturbations," *IEEE Trans. on Information Forensics and Security*, Vol. 17, pp. 1757—1770, 2022.
- P. R. Johnstone and P. Moulin, "Faster Subgradient Methods for Functions with Hölderian Growth," *Mathematical Programming*, Vol. 180, pp. 418—450, 2020.
- R. Trabelsi, J. Varadarajan, L. Zhang, I. Jabri, Y. Pei, F. Smach, A. Bouallegue, and P. Moulin, "Understanding the Dynamics of Social Interactions: A Multi-Modal Multi-View Approach" *ACM Transactions on Multimedia Computing, Communications, and Applications*, Feb. 2019.
- R. W. Yeung, C. Chen, Q. Chen, and P. Moulin, "On Characterizations of Markov Random Fields and Subfields," *IEEE Trans. Information Theory*, Aug. 2018.
- P. Moulin, "The Log-Volume of Optimal Codes on Memoryless Channels, Asymptotically Within a Few Nats," *IEEE Trans. Information Theory*, Vol. 63, pp. 2278—2313, April 2017.
- P. R. Johnstone and P. Moulin, "Local and global convergence of a general inertial proximal splitting scheme for minimizing composite functions," *Computational Optimization and Applications*, Feb. 2017.
- J. Lu, G. Wang and P. Moulin, "Localized Multi-Feature Metric Learning for Image Set Recognition," *IEEE Trans. Circ. Syst. Video Technology*, Vol. 26, No. 3, pp. 529—540, March 2016.
- B. Ni, V. R. Paramathayalan, T. Li, and P. Moulin, "Multiple Granularity Modeling: A Coarse-to-Fine Framework for Fine-grained Action Analysis," *International Journal on Computer Vision (IJCV)*, March 2016.
- H. Yu and P. Moulin, "SNR Maximization Hashing," *IEEE Trans. Information Forensics and Security*, Vol. 10, No. 9, pp. 1927—1938, Sep. 2015.
- B. Ni, G. Wang, and P. Moulin, "Order Preserving Sparse Coding" *IEEE Trans. on Pattern Recognition and Machine Intelligence*, Aug. 2015, Vol. 37, No. 8, pp. 1615—1628.
- H. Yu and P. Moulin, "Regularized Adaboost Learning for Identification of Time-Varying Content," *IEEE Trans. Information Forensics and Security*, Vol. 9, No. 10, pp. 1606—1616, Oct. 2014.
- Y.-W. Huang and P. Moulin, "On the Fingerprinting Capacity Games for Arbitrary Alphabets and Their Asymptotics," *IEEE Trans. Information Forensics and Security*, Vol. 9, No. 9, pp. 1477—1490, Sep. 2014.
- B. Ni, P. Moulin and S. Yan, "Pose Adaptive Motion Feature Pooling for Human Action Analysis," *International Journal on Computer Vision (IJCV)*, July 2014.
- J. Lu, G. Wang and P. Moulin, "Human Identity and Gender Recognition from Gait Sequences with Arbitrary Walking Directions," *IEEE Trans. on Information Forensics and Security*, Vol. 9, No. 1, pp. 41—51, Jan. 2014.
- B. Ni, Y. Pei, S. Yan, and P. Moulin, "Multi-Level Depth and Image Fusion for Human Activity Detection," *IEEE Transactions on System, Man and Cybernetics, Part B (TSMC-B)*, Vol. 43, No. 5, pp. 1383—1394, 2012.
- Y.-W. Huang and P. Moulin, "On the Saddle-point Solution and the Large-Coalition Be-

havior of Fingerprinting Games,” *IEEE Trans. on Information Forensics and Security*, Vol. 7, No. 1, pp. 160—175, 2012.

S. Sadasivam, P. Moulin and T. P. Coleman, “A Message Passing Approach to Combating Desynchronization Attacks,” *IEEE Trans. on Information Forensics and Security*, Vol. 6, pp. 894—905, 2011.

J.-F. Jourdas and P. Moulin, “High-Rate Random-Like Fingerprinting Codes with Linear Decoding Complexity,” *IEEE Transactions on Information Forensics and Security*, Vol. 4, No. 4, Dec. 2009.

N. Kiyavash, P. Moulin and T. Kalker, “Regular Simplex Fingerprints and Their Optimality Properties,” *IEEE Transactions on Information Forensics and Security*, Vol. 4, No. 3, pp. 318—329, Sep. 2009.

N. Kiyavash and P. Moulin, “Performance of Orthogonal Fingerprints Under Worst-Case Noise Attacks,” *IEEE Transactions on Information Forensics and Security*, Vol. 4, No. 3, pp. 293—301, Sep. 2009.

M. El Choubassi and P. Moulin, “On Reliability and Security of Randomized Detectors Against Sensitivity Analysis Attacks,” *IEEE Trans. Information Forensics and Security*, Vol. 4, No. 3, pp. 273—283, Sep. 2009.

S. Sadasivam and P. Moulin, “On Estimation Accuracy of Desynchronization Attack Channel Parameters,” *IEEE Transactions on Information Forensics and Security*, Vol. 4, No. 3, pp. 284—292, Sep. 2009.

P. Moulin, “A Neyman-Pearson Approach to Universal Erasure and List Decoding,” *IEEE Trans. Information Theory*, Oct. 2009.

Y. Wang and P. Moulin, “Perfectly Secure Steganography: Capacity, Error Exponents, and Code Constructions,” *IEEE Transactions on Information Theory*, Special Issue on Security, Vol. 54, No. 6, pp. 2706—2722, June 2008.

M. El Choubassi and P. Moulin, “Noniterative Algorithms for Sensitivity Analysis Attacks,” *IEEE Trans. Information Forensics and Security*, Vol. 2, No.3, pp. 113—126, June 2007.

P. Moulin and Y. Wang, “Capacity and Random-Coding Exponents for Channel Coding with Side Information,” *IEEE Trans. on Information Theory*, Vol. 53, No. 4, pp. 1326—1347, Apr. 2007.

Y. Wang and P. Moulin, “Optimized Feature Extraction for Learning-Based Image Steganalysis,” *IEEE Trans. Information Forensics and Security*, Vol. 2, No. 1, pp. 31—45, March 2007.

P. Moulin, “Signal Transmission with Known-Interference Cancellation,” *IEEE Signal Processing Magazine*, Lecture Notes, Vol. 24, No. 1, pp. 134—136, Jan. 2007.

J. C. Ye, P. Moulin and Y. Bresler, “Asymptotic Global Confidence Regions for Parametric 3-D Shape Estimation,” *IEEE Trans. Image Processing*, Vol. 15, No. 10, pp. 2904—2919, Oct. 2006.

P. Moulin and A. K. Goteti, “Block QIM Watermarking Games,” *IEEE Trans. Information Forensics and Security*, Vol. 1, No. 3, pp. 293—310, Sep. 2006.

S. Jana and P. Moulin, “Optimality of KLT for Encoding Gaussian Vector-Scale Mixtures: Application to Reconstruction, Estimation and Classification,” *IEEE Trans. on Informa-*

tion Theory, Vol. 52, No. 9, pp. 4049–4067, Sep. 2006.

T. Liu, P. Moulin and R. Koetter, “On Error Exponents of Modulo Lattice Additive Noise Channels,” *IEEE Trans. on Information Theory*, Vol. 52, No. 2, pp. 454–471, Feb. 2006.

P. Moulin and R. Koetter, “Data-Hiding Codes,” invited tutorial paper, *Proc. IEEE*, Vol. 93, No. 12, pp. 2083–2127, Dec. 2005.

P. Ishwar and P. Moulin, “On the Existence and Characterization of the Maxent Distribution Under General Moment Inequality Constraints,” *IEEE Trans. on Information Theory*, Vol. 51, No. 9, pp. 3322–3333, Sep. 2005.

J. L. Cannons and P. Moulin, “Design and Statistical Analysis of a Hash-Aided Image Watermarking System,” *IEEE Trans. on Image Processing*, Vol. 13, No. 10, pp. 1393–1408, Oct. 2004.

P. Moulin and M. K. Mıhçak, “The Parallel-Gaussian Watermarking Game,” *IEEE Trans. on Information Theory*, Vol. 50, No. 2, pp. 272–289, Feb. 2004.

P. Moulin, “Comments on “Why Watermarking is Nonsense,” *IEEE Signal Processing Magazine*, Vol. 20, No. 6, pp. 57–59, Nov. 2003.

P. Moulin and A. Ivanović, “The Zero-Rate Spread-Spectrum Watermarking Game,” *IEEE Transactions on Signal Processing*, Vol. 51, No. 4, pp. 1098–1117, Apr. 2003.

P. Moulin and J. A. O’Sullivan, “Information-Theoretic Analysis of Information Hiding,” *IEEE Trans. on Information Theory*, Vol. 49, No. 3, pp. 563–593, March 2003.

P. Ishwar and P. Moulin, “On the Equivalence Between Set-Theoretic and Maxent MAP Estimation,” *IEEE Trans. on Signal Processing*, Vol. 51, No. 3, pp. 698–713, March 2003.

J. C. Ye, Y. Bresler and P. Moulin, “Cramer-Rao Bounds for Parametric Shape Estimation in Inverse Problems,” *IEEE Trans. on Image Processing*, Vol. 12, No. 1, pp. 71–84, Jan. 2003.

J. C. Ye, Y. Bresler and P. Moulin, “A Self-Referencing Level-Set Method for Image Reconstruction from Sparse Fourier Samples,” invited paper, *Int. J. of Computer Vision*, special issue on level-set methods, Dec. 2002.

P. Moulin and M. K. Mıhçak, “A Framework for Evaluating the Data-Hiding Capacity of Image Sources,” *IEEE Trans. on Image Processing*, Vol. 11, No. 9, pp. 1029–1042, Sep. 2002.

A. Jain, P. Moulin, M. I. Miller and K. Ramchandran, “Information-Theoretic Bounds on Target Recognition Performance Based on Degraded Image Data,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 9, pp. 1153–1166, Sep. 2002.

J. Liu and P. Moulin, “Information-Theoretic Analysis of Interscale and Intrascale Dependencies Between Image Wavelet Coefficients,” *IEEE Trans. on Image Processing*, Vol. 10, No. 10, pp. 1647–1658, Nov. 2001.

P. Moulin, invited discussion of “Regularization of Wavelet Approximations,” by A. An-toniadis and J. Fan, *Journal of the American Statistical Association*, Vol. 96, No. 455, pp. 959–960, Sep. 2001.

P. Moulin, “The Role of Information Theory in Watermarking and Its Application to Image Watermarking,” invited paper, *Signal Processing*, Vol. 81, No. 6, pp. 1121–1139, June 2001.

J. Liu and P. Moulin, "Complexity-Regularized Image Denoising," *IEEE Trans. on Image Processing*, Vol. 10, No. 6, pp. 841—851, June 2001.

J. C. Ye, Y. Bresler and P. Moulin, "Cramer-Rao Bounds for Parametric Boundaries of Targets in Inverse Scattering Problems," *IEEE Trans. on Antennas and Propagation*, May 2001.

M. K. Mıhçak, P. Moulin, M. Anitescu, and K. Ramchandran, "Rate-Distortion-Optimal Subband Coding Without Perfect Reconstruction Constraints," *IEEE Trans. on Signal Processing*, Vol. 49, No. 3, pp. 542—557, Mar. 2001.

P. Ishwar and P. Moulin, "On Spatial Adaptation of Motion Field Smoothness in Video Coding," *IEEE Trans. Circ. Syst. Video Tech.*, Vol. 10, No. 6, pp. 980–989, Sep. 2000.

P. Moulin and J. Liu, "Statistical Imaging and Complexity Regularization," *IEEE Trans. on Information Theory*, Special issue on information-theoretic imaging, Vol. 46, No. 5, pp. 1762—1777, Aug. 2000.

J. C. Ye, Y. Bresler and P. Moulin, "Asymptotic Global Confidence Regions in Parametric Shape Estimation Problems," *IEEE Trans. on Information Theory*, Special issue on information-theoretic imaging, Vol. 46, No. 5, pp. 1881–1895, Aug. 2000.

P. Moulin, M. Anitescu, and K. Ramchandran, "Theory of Rate-Distortion-Optimal, Constrained Filter Banks — Application to FIR and IIR Biorthogonal Designs," *IEEE Trans. on Signal Processing*, Vol. 48, No. 4, pp. 1120–1132, April 2000.

M. K. Mıhçak, I. Kozintsev, K. Ramchandran and P. Moulin, "Low-Complexity Image Denoising Based on Statistical Modeling of Wavelet Coefficients," *IEEE Signal Processing Letters*, Vol. 6, No. 12, pp. 300—303, Dec. 1999.

R. Krishnamurthy, J. W. Woods and P. Moulin, "Frame Interpolation and Bidirectional Prediction of Video Using Compactly-Encoded Optical Flow Fields and Label Fields," *IEEE Trans. Circ. Syst. Video Tech.*, Vol. 9, No. 5, pp. 713—726, Aug. 1999.

P. Moulin and J. Liu, "Analysis of Multiresolution Image Denoising Schemes Using Generalized-Gaussian and Complexity Priors," *IEEE Trans. on Info. Theory*, Special Issue on Multi-scale Analysis, Vol. 45, No. 3, pp. 909-919, Apr. 1999.

V. Pavlovic, P. Moulin and K. Ramchandran, "An Integrated Framework for Adaptive Subband Image Coding," *IEEE Trans. on Signal Processing*, Vol. 47, No. 4, pp. 1024-1038, Apr. 1999.

P. Moulin and M. K. Mıhçak, "Theory and Design of Signal-Adapted FIR Paraunitary Filter Banks," *IEEE Trans. on Signal Processing*, Special Issue on Wavelets and Filter Banks, Vol. 46, No. 4, pp. 920—929, Apr. 1998.

P. Moulin, R. Krishnamurthy, "Multiscale Modeling and Estimation of Motion Fields for Video Coding," *IEEE Trans. on Image Processing*, Vol. 6, No. 12, pp. 1606—1620, Dec. 1997.

P. Moulin, M. Anitescu, K.O. Kortanek and F. Potra, "The Role of Linear Semi-Infinite Programming in Signal-Adapted QMF Bank Design," *IEEE Trans. on Signal Processing*, Vol. 45, No. 9, pp. 2160—2174, Sep. 1997.

P. Moulin, "A Multiscale Relaxation Technique for SNR Maximization in Nonorthogonal Subband Coding," *IEEE Trans. on Image Processing*, Vol. 4, No. 9, pp. 1269—1281, Sep. 1995.

P. Moulin, A.T. Ogielski, G. Lilienfeld and J.W. Woods: "Video Signal Processing and Coding on Data-Parallel Computers," *Digital Signal Processing: A Review Journal*, Vol. 5, No. 2, pp. 118–129, Apr. 1995.

P. Moulin, invited discussion of "Wavelet Shrinkage: Asymptotia?" by D. Donoho and I. Johnstone, *Journal of the Royal Statistical Society B*, Vol. 57, No. 1, 1995.

P. Moulin, "Wavelet Thresholding Techniques for Power Spectrum Estimation," *IEEE Trans. on Signal Processing*, Vol. 42, No. 11, pp. 3126–3136, Nov. 1994.

P. Moulin, "A Wavelet Regularization Method for Diffuse Radar-Target Imaging and Speckle-Noise Reduction," *Journal of Mathematical Imaging and Vision*, Special Issue on Wavelets, Vol. 3, No. 1, pp. 123–134, 1993.

J. A. O'Sullivan, P. Moulin, and D. L. Snyder, "An Application of Splines to Maximum Likelihood Radar Imaging," *International Journal of Imaging Systems and Technology*, Vol. 4, pp. 256–264, 1993.

P. Moulin, J. A. O'Sullivan, and D. L. Snyder, "A Method of Sieves for Multiresolution Spectrum Estimation and Radar Imaging," *IEEE Trans. on Information Theory*, Special Issue on Wavelets and Multiresolution Analysis, pp. 801–813, Mar. 1992.

Book P. Moulin and V. Veeravalli, *Statistical Inference for Engineers and Data Scientists*, Cambridge University Press, 2019.

Book Chapters

B. Ni, G. Wang, and P. Moulin, "RGBD-HuDaAct: A color-depth video database for human daily activity recognition," *Consumer Depth Cameras for Computer Vision: Advances in Computer Vision and Pattern Recognition*, pp 193-208, Springer-Verlag, 2013.

P. Moulin, "Information-Hiding Games," *Springer-Verlag Lecture Notes in Computer Sciences*, Vol. 2613, 2003.

P. Moulin, "Multiscale Image Decompositions and Wavelets," *Handbook of Image and Video Processing*, A. C. Bovik, Ed., Academic Press, 2000.

K.O. Kortanek and P. Moulin, "Using Semi-Infinite Programming in Orthogonal Wavelet Filter Design," *Semi-Infinite Programming and Its Applications*, Kluwer Academic series on Nonconvex Optimization and its Applications, 1998.

P. Moulin, J. A. O'Sullivan, and D. L. Snyder, "A Sieve-Constrained Maximum-Likelihood Method for Target Imaging," in *Radar and Sonar II*, Springer-Verlag, The IMA Volumes in Mathematics and Its Applications, Vol. 39, Eds. F. Grunbaum et al., pp. 95–122, 1992.

P. Moulin, "Adaptive Multiresolution Image Restoration and Compression," in *IEEE Case Studies in Medical Instrument Design*, Eds. H. Troy Nagle and W.J. Tomkins, pp. 247–254, IEEE, New York, 1992.

Conference Papers

A. Rakin, Y. Wang, S. Aaron, T. Koike-Akino, P. Moulin and K. Pearsons, "Towards Universal Adversarial Examples and Defenses," *Proc. IEEE Information Theory Workshop*, Kanazawa, Japan, Oct. 2021.

Ye Wang, Shuchin Aeron, Adnan Rakin, Toshiaki Koike-Akino and P. Moulin, "Robust Machine Learning via Privacy/Rate-Distortion Theory," *Proc. IEEE Int. Symp. On Information Theory*, Melbourne, Australia, July 2021.

A. Goel and P. Moulin, "Locally Optimal Detection of Stochastic Targeted Universal Adversarial Perturbations," *Proc. ICASSP*, Toronto, Canada, May 2021.

F. Zhuang and P. Moulin, "Deep Semi-Supervised Metric Learning Via Identification of Manifold Memberships" *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, Canada, May 2021.

T. Jayashankar, P. Moulin, and J. Le Roux, "Detecting Audio Attacks on ASR Systems with Dropout Uncertainty," *Proc. Interspeech*, Oct. 2020.

F. Zhang and P. Moulin, "A New Variational Method for Deep Supervised Image Hashing," *Proc. ICASSP*, Barcelona, Spain, May 2020.

P. Moulin, "Game-theoretic design of universal adversarial perturbation detectors," presented at *Information Theory and Applications*, San Diego, CA, Feb. 2020.

T. Jayashankar, P. Moulin, T. Blu, C. Gilliam, LAP-Based Video Frame Interpolation, *Proc. ICIP 2019*, Taipei, Taiwan, Sep. 2019.

P. Moulin and A. Goel, "Random Ensemble of Locally Optimum Detectors for Detection of Adversarial Inputs," *IEEE GlobalSIP Conf., Signal Processing for Adversarial Machine Learning*, Anaheim, CA, Nov. 2018.

P. Moulin, "Gaussian Mixture Models for Detection of Adversarial Inputs," *Information Theory and Applications*, San Diego, CA, Feb. 2018.

P. Moulin and A. Goel, "Locally Optimal Detection of Adversarial Inputs to Image Classifiers," *Proc. ICME*, Hong Kong, July 2017.

R. W. Yeung, C. Chen, Q. Chen, and P. Moulin, Information-Theoretic Characterizations of Markov Random Field and Subfield, *Proc. ISIT*, Aachen, Germany, June 2017.

P. Moulin, "Source Coding with Distortion Profile Constraints", *Proc. ISIT*, Aachen, Germany, June 2017.

P. Moulin, "Lower Bounds on Rate of Fixed-Length Source Codes under Average- and f-Fidelity Constraints," *Proc. ISIT*, Aachen, Germany, June 2017.

Zhang Le, J. Varadarajan, P. N. Suganthan, N. Ahuja, and P. Moulin, "Robust Visual Tracking Using Incremental Oblique Random Forest," *CVPR*, June 2017.

J. Varadarajan, R. Subramanian, N. Ahuja, P. Moulin, J.-M. Odobez, "Active Online Anomaly Detection Using Dirichlet Process Mixture Model and Gaussian Process Classification," *Proc. Winter Conference on Applications of Computer Vision*, Santa Rosa, CA, March 2017.

P. R. Johnstone and P. Moulin, "Convergence Rates of Inertial Splitting Schemes for Nonconvex Composite Optimization," *Proc. ICASSP*, New Orleans, LA, March 2017.

P. Moulin, "Refined asymptotic analysis of source coding under excess-distortion constraint," *ITA*, San Diego, CA, Feb. 2017.

P. Moulin, "Asymptotically Achievable Error Probabilities for Multiple Hypothesis Testing," *Proc. IEEE Int. Symp. Information Theory*, Barcelona, Spain, July 2016.

P. Moulin, "Strong Large Deviations for Composite Multiple Hypothesis Testing," *ITA*, San Diego, CA, Feb. 2016.

H. Yu and P. Moulin, "Multi-Feature Hashing Based on SNR Maximization," *Proc. ICIP*, Quebec City, Canada, Sep. 2015.

T. Blu, P. Moulin, and C. Gilliam, "Approximation Order of the LAP Optical Flow Algorithm," *Proc. ICIP*, Quebec City, Canada, Sep. 2015.

P. Moulin, "Coding Redundancy of Finite-Blocklength Universal Channel Coding," *Proc. IEEE Int. Symp. Information Theory*, Hong Kong, June 2015.

P. Moulin and P. R. Johnstone, "Strong Large Deviations for GLRT and Variants in Composite Hypothesis Testing," *Proc. IEEE Int. Symp. Information Theory*, Hong Kong, June 2015.

J. Lee, M. Raginsky, and P. Moulin, "On MMSE Estimation from Quantized Observations in the Nonasymptotic Regime," *Proc. IEEE Int. Symp. Information Theory*, Hong Kong, June 2015.

B. Ni and P. Moulin, "Motion Part Regularization: Improving Action Recognition via Trajectory Selection," *Computer Vision and Pattern Recognition (CVPR)*, June 2015.

J. Lu, G. Wang, W. Deng, P. Moulin, and J. Zhou, "Multi-Manifold Deep Learning for Image Set Classification," *Proc. CVPR*, June 2015.

V. E. Liong, J. Lu, G. Wang, P. Moulin, and J. Zhou, "Deep Hashing for Compact Binary Codes Learning," *Proc. CVPR*, June 2015.

H. Yu and P. Moulin, "SNR Maximization Hashing," *Proc. ICASSP*, Brisbane, Australia, Apr. 2015.

P. R. Johnstone and P. Moulin, "Convergence of an Inertial Proximal Method for L_1 Regularized Least Squares," *Proc. ICASSP*, Brisbane, Australia, Apr. 2015.

S. D. Chen and P. Moulin, "A Classification Centric Quantizer for Efficient Encoding of Predictive Feature Errors," *Proc. Asilomar Conf.*, Monterrey, CA, Nov. 2014.

J. Lu, G. Wang, W. Deng, and P. Moulin, "Simultaneous feature learning and dictionary learning for image set base face recognition," *ECCV*, Lecture Notes in Computer Science Vol. 8689, pp. 265—280, Zurich, Switzerland, Sep. 2014.

Y. Zhou, B. Ni, S. Yan, P. Moulin, and Q. Tian, "Pipelining Localized Semantic Features for Fine-Grained Action Recognition," *ECCV*, Lecture Notes in Computer Science Vol. 8692, pp. 481—496, Zurich, Switzerland, Sep. 2014.

V. Tan and P. Moulin, "Second- and Higher-Order Asymptotics For Erasure and List Decoding," *Proc. ISIT*, Honolulu, HI, pp. 1887—1891, July 2014.

Y.-W. Huang and P. Moulin, "Strong Large Deviations for Composite Hypothesis Testing," *Proc. ISIT*, Honolulu, HI, pp. 556—560, July 2014.

B. Ni, T. Li, and P. Moulin, "Beta Process Multiple Kernel Learning," *CVPR*, Columbus, OH, pp. 963—970, June 2014.

B. Ni, V. R. Paramathayalan and P. Moulin, "Multiple Granularity Analysis for Fine-Grained Action Detection," *CVPR*, Columbus, OH, pp. 756—763, June 2014.

S. D. Chen and P. Moulin, "A Two-Part Predictive Coder for Multitask Signal Compression," *Proc. ICASSP*, Florence, Italy, pp. 2035—2039, May 2014.

R. Naini and P. Moulin, "Fingerprint Information Maximization for Content Identification," *Proc. ICASSP*, Florence, Italy, pp. 3809–3813, May 2014.

P. Moulin and P. R. Johnstone, "Kullback-Leibler Divergence and the Central Limit Theorem," *Proc. ITA*, San Diego, CA, Feb. 2014.

J. Lu, G. Wang, and P. Moulin, "Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning," *IEEE International Conference on Computer Vision (ICCV)*, pp. 329–336, 2013.

B. Ni and P. Moulin, "Manipulation Pattern Discovery: A Nonparametric Bayesian Approach," *ICCV*, 2013.

P. Moulin, "Asymptotic Neyman-Pearson Games for Converse to the Channel Coding Theorem," *Proc. ISIT*, Istanbul, July 2013.

P. R. Johnstone, A. Emad, P. Moulin, and O. Milenkovic, "RFIT: A New Algorithm for Matrix Completion," *Proc. SPARS*, Lausanne, Switzerland, July 2013.

D. Sungatullina, J. Lu, G. Wang, and P. Moulin, "Multiview discriminative learning for age-invariant face recognition," *IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pp. 1–6, Shanghai, China, Apr. 2013.

H. Yu and P. Moulin, "Regularized Adaboost for Content identification," *Proc. ICASSP*, Vancouver, Canada, May 2013.

H. Yu, P. Moulin and S. Roy, "RGB-D Video Content Identification," *Proc. ICASSP*, Vancouver, Canada, May 2013.

D. Sungatullina, J. Lu, G. Wang, and P. Moulin, "Multiview Discriminative Learning for Age-Invariant Face Recognition," *Proc. 10th IEEE International Conference on Automatic Face and Gesture Recognition*, Shanghai, China, Apr. 2013.

P. Moulin, "A New Metaconverse and Outer Region for Finite-Blocklength MACs," *Proc. Information Theory and Applications (ITA) Workshop*, San Diego, CA, Feb. 2013.

A. Singh, N. Ahuja, P. Moulin, "Online learning with kernels: Overcoming the growing sum problem," *Proc. IEEE Machine Learning for Signal Processing (MLSP)*, Santander, Spain, pp. 449-454, Sep. 2012.

B. Ni, M. Xu, J. Tang, S. Yan, and P. Moulin, "Omni-range spatial contexts for visual classification," *Proc. CVPR*, pp. 3514-3521, Providence, RI, June 2012.

B. Ni, P. Moulin, S. Yan, "Order-Preserving Sparse Coding for Sequence Classification," *Proc. ECCV* pp. 173-187, Providence, RI, June 2012.

B. Ni, N. C. Dat, P. Moulin, "RGBD-camera based get-up event detection for hospital fall prevention," *Proc. ICASSP*, pp. 1405-1408, Kyoto, Japan, March 2012.

R. Dubey, B. Ni, and P. Moulin, "A Depth Camera Based Fall Recognition System for the Elderly," *Proc. 9th Int. Conf. Image Analysis and Recognition (ICIAR)*, LNCS Vol. 7325, pp. 106-113, Aveiro, Portugal, June 2012.

P. Moulin, "The log-volume of optimal constant-composition codes for memoryless channels, within $O(1)$ bits," *Proc. ISIT*, pp. 826-830, Cambridge, MA, July 2012.

Y.-W. Huang and P. Moulin, "Finite blocklength coding for multiple access channels," *Proc. ISIT*, pp. 831-835, Cambridge, MA, July 2012.

Y.-W. Huang and P. Moulin, "On fingerprinting capacity games for arbitrary alphabets and their asymptotics," *Proc. ISIT*, pp. 2571-2575, Cambridge, MA, July 2012.

R. Naini and P. Moulin, "Real Adaboost for content identification," *Proc. IEEE Stat. Signal Processing Workshop*, Ann Arbor, MI, Aug. 2012.

R. Naini and P. Moulin, "Model-based decoding metrics for content identification," *Proc. ICASSP*, pp. 1829-1832, Kyoto, Japan, March 2012.

P. Moulin, "The log-volume of optimal codes for memoryless channels, Up to a Few Nats," *Proc. Information Theory and Applications Workshop*, San Diego, CA, Feb. 2012.

B. Ni, G. Wang, and P. Moulin, "RGBD-HuDaAct: A color-depth video database for human daily activity recognition," *Proc. 1st IEEE workshop on Consumer Depth Cameras for Computer Vision*, held in conjunction with *Int. Conf. on Computer Vision (ICCV) Workshop*, pp. 1147-1153, Barcelona, Nov. 2011.

S. Sadasivam and P. Moulin, "A Universal Divergence-Rate Estimator for Steganalysis in Timing Channels," *Proc. 2nd Workshop on Information Security and Forensics*, Seattle, WA, Dec. 2010.

Y.-W. Huang and P. Moulin, "On the Saddle-point Solution and the Large-Coalition Behavior of Fingerprinting Games," *Proc. 2nd Workshop on Information Security and Forensics*, Seattle, WA, Dec. 2010.

R. Naini and P. Moulin, "Interference Management through Mobile Relays in Ad Hoc Networks," *Proc. ASILOMAR Conf.*, Pacific Grove, CA, Nov. 2010.

G. Gigaud and P. Moulin, "A Geometrically Resilient SURF-Based Image Fingerprinting Scheme," *Proc. IEEE Int. Conf. Image Processing (ICIP)*, Hong Kong, Sep. 2010.

G. Gigaud and P. Moulin, "Traitor Tracing Aided by Compressed SURF Features," *Proc. Conf. Info. Sci. and Sys.*, Princeton, NJ, March 2010.

P. Moulin, "Statistical Modeling and Analysis of Content Identification and Retrieval," *Proc. Information Theory and Applications Workshop*, San Diego, CA, Feb. 2010.

Y.-W. Huang and P. Moulin, "Capacity-Achieving Fingerprint Decoding," *Proc. 1st Workshop on Information Security and Forensics*, London, UK, Dec. 2009.

S. Sadasivam and P. Moulin, "Combating Desynchronization Attacks on Blind Watermarking Systems: A Message Passing Approach," *Proc. ICIP*, Cairo, Egypt, Nov. 2009.

I. Ezzeddine and P. Moulin, *Proc. IEEE Information Theory Workshop*, Taormina, Italy, Oct. 2009.

S. D. Chen, V. Monga, and P. Moulin, "Meta-classifiers For Multimodal Document Classification," *Proc. IEEE Multimedia Signal Processing Workshop*, Rio de Janeiro, Brazil, Oct. 2009.

Y.-W. Huang and P. Moulin, "Saddlepoint Solution of the Fingerprinting Capacity Game under the Marking Assumption," *Proc. Int. Symp. Information Theory*, Seoul, Korea, July 2009.

P. Moulin, "A Strong Converse for the Gel'fand-Pinsker Channel," *Proc. Int. Symp. Information Theory*, Seoul, Korea, July 2009.

P. Moulin, "Optimal Gaussian Fingerprint Decoders" *Proc. ICASSP*, Taipei, Taiwan, Apr. 2009.

P. Moulin and Y. Wang, "Information-Theoretic Analysis of Spherical Fingerprinting," *Proc. ITA Workshop*, San Diego, CA, Feb. 2009.

M. El Choubassi and P. Moulin, "Joint Estimation-Detection Games for Sensitivity Analysis Attacks," *Proc. SPIE Conf.*, San Jose, CA, Jan. 2009.

P. Moulin, "A Neyman-Pearson Approach to Universal Erasure and List Decoding," *Proc. IEEE Symp. Information Theory*, pp. 61—65, July 2008.

P. Moulin, "Universal Fingerprinting: Capacity and Random-Coding Exponents," *Proc. IEEE Symp. Information Theory*, pp. 220—224, July 2008.

J.-F. Jourdas and P. Moulin, "Towards Optimal Design of Multimedia Fingerprinting Codes," *Proc. ICASSP*, Las Vegas, NV, pp. 1649—1652, April 2008.

J.-F. Jourdas and P. Moulin, "A Low-Rate Fingerprinting Code And Its Application To Blind Image Fingerprinting," *Proc. SPIE*, San Jose, CA, Jan. 2008.

J.-F. Jourdas and P. Moulin, "A High-Rate Fingerprinting Code," *Proc. SPIE*, San Jose, CA, Jan. 2008.

S. Sadasivam, P. Moulin, and R. Koetter, "Graphical Models for Desynchronization-Resilient Watermark Decoding," *Proc. IEEE Int. Conf. on Image Proc.*, San Antonio, TX, pp. V.477—480, Sep. 2007.

M. El Choubassi and P. Moulin, "Sensitivity Analysis Attacks Against Randomized Detectors," *Proc. IEEE Int. Conf. on Image Proc.*, San Antonio, TX, pp. II.129—132, Sep. 2007.

N. Kiyavash and P. Moulin, "Digital Fingerprinting: on Colluders Worst-Case Noise," *IEEE Workshop on Statistical Signal Processing*, Madison, WI, July 2007.

P. Moulin and N. Kiyavash, "Expurgated Gaussian Fingerprinting Codes," *Proc. IEEE Int. Symp. on Information Theory*, Nice, France, June 2007.

P. Moulin and N. Kiyavash, "Performance of Random Fingerprinting Codes Under Arbitrary Nonlinear Attacks," *Proc. ICASSP*, Honolulu, HI, pp. II.157—160, April 2007.

N. Kiyavash and P. Moulin, "Sphere Packing Lower Bound on Fingerprinting Error Probability," *Proc. SPIE*, San Jose, CA, Jan. 2007.

Y. Wang and P. Moulin, "Capacity and Optimal Collusion Attack Channels for Gaussian Fingerprinting Games," *Proc. SPIE*, San Jose, CA, Jan. 2007.

S. Sadasivam and P. Moulin, "Cramer-Rao Bound on Watermark Desynchronization Parameter Accuracy," *Proc. SPIE*, San Jose, CA, Jan. 2007.

L. Perez-Freire, P. Moulin, and F. Perez-Gonzalez, "Security of spread-spectrum-based data hiding," in *Security, Steganography, and Watermarking of Multimedia Contents IX*, Edward J. Delp III and P. W. Wong, Eds., vol. 6505. San Jose, California, USA: SPIE, January 2007.

P. Moulin, "On the Optimal Structure of Watermark Decoders Under Geometric Attacks," *Proc. IEEE Int. Conf. on Image Proc.*, Atlanta, GA, Oct. 2006.

P. Moulin, "Universal Decoding of Watermarks Under Geometric Attacks" *Proc. IEEE Int. Symp. on Information Theory*, July 2006.

Y. Wang and P. Moulin, "Capacity and Random-Coding Error Exponent for Public Fin-

- gerprinting Game,” *Proc. IEEE Int. Symp. on Information Theory*, July 2006.
- S. Ray, P. Moulin and M. Médard, “On Jamming in the Wideband Regime,” *Proc. IEEE Int. Symp. on Information Theory*, July 2006.
- S. Ray, P. Moulin and M. Médard, “On Optimal Signaling and Jamming Strategies in Wideband Fading Channels,” *Proc. IEEE Conf. on Signal Processing Applications for Wireless Communications (SPAWC)*, June-July 2006.
- N. Kiyavash and P. Moulin, “On Optimal Collusion Strategies for Fingerprinting,” *Proc. ICASSP*, Toulouse, France, May 2006.
- N. Kiyavash and P. Moulin, “A Framework for Optimizing Nonlinear Collusion Attacks on Fingerprinting Systems,” *Proc. CISS*, Princeton, NJ, March 2006.
- P. Moulin and R. Koetter, “A Framework for Design of Good Watermark Identification Codes,” *Proc. SPIE*, San Jose, CA, Jan. 2006.
- M. El Choubassi and P. Moulin, “On the fundamental tradeoff between watermark detection performance and robustness against sensitivity analysis attacks,” *Proc. SPIE*, San Jose, CA, Jan. 2006.
- Y. Wang and P. Moulin, “Statistical modeling and steganalysis of DFT-based image steganography,” *Proc. SPIE*, San Jose, CA, Jan. 2006.
- N. Kiyavash and P. Moulin, “Regular Simplex Fingerprints and Their Optimality Properties,” *Proc. Int. Workshop on Digital Watermarking*, Siena, Italy, Sep. 2005.
- P. Moulin and Y. Wang, “Improved QIM Strategies for Gaussian Watermarking,” *Proc. Int. Workshop on Digital Watermarking*, Siena, Italy, Sep. 2005.
- P. Moulin and A. K. Goteti, “Minmax Strategies for QIM Watermarking Subject to Attacks with Memory,” *Proc. IEEE ICIP*, Sep. 2005.
- P. Moulin and Y. Wang, “On Achievable Error Exponents for Watermarking,” *Proc. SPIE Conf.*, San Jose, CA, Jan. 2005.
- M. El Choubassi and P. Moulin, “A New Sensitivity Analysis Attack,” *Proc. SPIE Conf.*, San Jose, CA, Jan. 2005.
- A. K. Goteti and P. Moulin, “QIM Watermarking Games,” *Proc. ICIP*, Singapore, Oct. 2004.
- P. Moulin and A. Briassouli, “A Stochastic QIM Algorithm for Robust, Undetectable Image Watermarking,” invited paper, *Proc. ICIP*, Singapore, Oct. 2004.
- T. Liu, P. Moulin and R. Koetter, “On Error Exponents of Nested Lattice Codes for the AWGN Channel,” *Proc. Information Theory Workshop*, San Antonio, TX, Oct. 2004.
- P. Moulin and Y. Wang, “Error Exponents for Channel Coding with Side Information,” *Proc. Information Theory Workshop*, San Antonio, TX, Oct. 2004.
- P. Moulin, A. K. Goteti and R. Koetter, “Optimal Sparse QIM Codes for Zero-Rate Blind Watermarking,” *Proc. ICASSP*, Montreal, Canada, May 2004.
- A. K. Goteti and P. Moulin, “Two Private, Perceptual Data-Hiding Games,” *Proc. ICASSP*, Montreal, Canada, May 2004.
- P. Moulin and Y. Wang, “New Results on Steganographic Capacity,” *Proc. Conf. on Info.*

Syst. and Sci., Princeton, NJ, March 2004.

Y. Wang and P. Moulin, "Steganalysis of Block-Structured Stegotext," *Proc. SPIE* Vol. 5306, San Jose, CA, Jan. 2004.

P. Moulin, "Embedded-Signal Design for Channel Parameter Estimation Part I: Linear Embedding," *Proc. IEEE Workshop on Statistical Signal Processing*, St Louis, MO, Sep. 2003.

P. Moulin, "Embedded-Signal Design for Channel Parameter Estimation Part II: Quantization Embedding," *Proc. IEEE Workshop on Statistical Signal Processing*, St Louis, MO, Sep. 2003.

Y. Wang and P. Moulin, "Steganalysis of Block-DCT Steganography," *Proc. IEEE Workshop on Statistical Signal Processing*, St Louis, MO, Sep. 2003.

J. Liu and P. Moulin, "On the Risk of Translation-Averaged Wavelet Estimators," *Proc. SPIE*, San Diego, Aug. 2003.

S. Jana and P. Moulin, "Optimal Transform Coders for Classification/Reconstruction with Joint Decoding at the Receiver," *Proc. IEEE Int. Symp. on Information Theory*, Yokohama, Japan, July 2003.

T. Liu and P. Moulin, "Error Exponents for Zero-Rate Watermarking Game with Squared-Error Constraints," *Proc. IEEE Int. Symp. on Information Theory*, Yokohama, Japan, July 2003.

T. Liu and P. Moulin, "Error Exponents for One-Bit Watermarking," *IEEE Proc. ICASSP*, Hong Kong, April 2003.

A. Briassouli and P. Moulin, "Detection-Theoretic Analysis of Warping Attacks in Spread-Spectrum Watermarking," *IEEE Proc. ICASSP*, Hong Kong, April 2003.

S. Jana and P. Moulin, "Optimal Transform Coding of Gaussian Mixtures for Joint Classification/Reconstruction," *Proc. Data Compression Conference*, Snowbird, UT, March 2003.

N. Schmid, Y. Bresler and P. Moulin, "Complexity Regularized Shape Estimation from Noisy Fourier Data," *Proc. IEEE Int. Conf. on Image Processing*, Rochester, NY, 2002.

J. C. Ye, Y. Bresler and P. Moulin, "Cramer-Rao Bounds for Parametric Shape Estimation in Inverse Problems," *Proc. IEEE Int. Conf. on Image Processing*, Rochester, NY, 2002.

A. Ivanović and P. Moulin, "Game-Theoretic Performance Analysis of Image Watermark Detectors," *Proc. IEEE Int. Conf. on Image Processing*, Rochester, NY, 2002.

M. K. Mıhçak and P. Moulin, "Information-Embedding Codes Matched to Local Gaussian Image Models," *Proc. IEEE Int. Conf. on Image Processing*, Rochester, NY, 2002.

P. Moulin, A. Briassouli and H. Malvar, "Detection-Theoretic Analysis of Desynchronization Attacks in Watermarking," *Proc. DSP'02*, Santorini, Greece, July 2002.

P. Moulin and M. T. Orchard, "Undergraduate Education in Image and Video Processing," *Proc. Int. Conf. on Ac., Sp., and Signal Processing*, Orlando, FL, May 2002.

S. C. Herman and P. Moulin, "A Particle Filtering Approach to Joint Radar Tracking and Automatic Target Recognition," *Proc. IEEE Aerospace Conference*, Big Sky, Montana, March 2002.

P. Moulin and A. Briassouli, "The Gaussian Fingerprinting Game," *Proc. CISS'02*, Princeton, NJ, March 2002.

P. Moulin and J. A. O'Sullivan, "Optimal Key Design in Information-Embedding Systems," *Proc. CISS'02*, Princeton, NJ, March 2002.

P. Moulin and A. Ivanović, "The Fisher Information Game for Optimal Design of Synchronization Patterns in Blind Watermarking," *Proc. IEEE Int. Conf. on Image Processing*, Thessaloniki, Greece, Oct. 2001.

P. Moulin and A. Ivanović, "Game-Theoretic Analysis of Watermark Detection," *Proc. IEEE Int. Conf. on Image Processing*, Thessaloniki, Greece, Oct. 2001.

J. C. Ye, Y. Bresler and P. Moulin, "A Self-Referencing Level-Set Method for Image Reconstruction from Sparse Fourier Samples," *Proc. IEEE Int. Conf. on Image Processing*, Thessaloniki, Greece, Oct. 2001.

J. Liu and P. Moulin, "Statistical Image Restoration Based on Adaptive Wavelet Models," *Proc. IEEE Int. Conf. on Image Processing*, Thessaloniki, Greece, Oct. 2001.

J. Liu and P. Moulin, "Approximation-Theoretic Analysis of Translation-Invariant Wavelet Denoising," *Proc. IEEE Int. Conf. on Image Processing*, Thessaloniki, Greece, Oct. 2001.

J. A. O' Sullivan and P. Moulin, "Some Properties of Optimal Information Hiding and Information Attacks," *Proc. 39th Allerton Conf.*, pp. 92-101, Monticello, IL, Oct. 3-5, 2001.

J. C. Ye, Y. Bresler and P. Moulin, "A Self-Referencing Level-Set Method for Image Reconstruction from Sparse Fourier Samples," *Proc. IEEE Int. Workshop on Variational and Level Set Methods*, Vancouver, B.C., July 2001.

P. Moulin and A. Ivanović, "The Watermark Selection Game," *Proc. Conf. on Info. Syst. and Sci.*, Baltimore, MD, March 2001.

P. Moulin, "The Parallel-Gaussian Watermarking Game," *Proc. Conf. on Info. Syst. and Sci.*, Baltimore, MD, March 2001.

J. Liu and P. Moulin, "Approximation-Theoretic Analysis of Translation-Invariant Wavelet Denoising," invited paper, *Proc. Conf. on Info. Syst. and Sci.*, Baltimore, MD, March 2001.

P. Moulin, M. K. Mıhçak and G.-I. Lin, "An Information-Theoretic Model for Image Watermarking," *IEEE Int. Conf. on Image Processing*, Vancouver, BC, Sep. 2000.

R. Venkatesan, M. Jakubowski, S.-M. William Koon and P. Moulin, "Robust Image Hashing," *IEEE Int. Conf. on Image Processing*, Vancouver, BC, Sep. 2000.

P. Ishwar and P. Moulin, "Shift-Invariant Image Denoising — A Multiple-Domain Maxent MAP Framework," *IEEE Int. Conf. on Image Processing*, Vancouver, BC, Sep. 2000.

J. Liu and P. Moulin, "Analysis of Interscale and Intracscale Dependencies Between Image Wavelet Coefficients," *IEEE Int. Conf. on Image Processing*, Vancouver, BC, Sep. 2000.

S. Jana and P. Moulin, "Optimal Design of Transform Coders and Quantizers for Image Classification," *IEEE Int. Conf. on Image Processing*, Vancouver, BC, Sep. 2000.

M. K. Mıhçak, A. F. Docimo, P. Moulin and K. Ramchandran, "Design and Analysis of a Forward-Adaptive Wavelet Image Coder," *IEEE Int. Conf. on Image Processing*, Vancouver, BC, Sep. 2000.

M. Kesal, M. K. Mihçak, R. Kötter and P. Moulin, "Iteratively Decodable Codes for Watermarking Applications," *Proc. 2nd Symposium on Turbo Codes and Related Topics*, pp. 113-116, Brest, France, Sep. 2000.

P. Ishwar and P. Moulin, "Image Modeling and Restoration – Information fusion, Set-theoretic methods, and the Maximum Entropy Principle," *Proc. MAXENT'00*, Paris, France, July 2000.

P. Moulin and J. A. O'Sullivan, "Information-Theoretic Analysis of Information Hiding," *Proc. IEEE Int. Symp. on Information Theory*, Sorrento, Italy, June 2000.

P. Moulin and J. Liu, "Statistical Imaging and Complexity Regularization," *Proc. IEEE Int. Symp. on Information Theory*, Sorrento, Italy, June 2000.

P. Moulin and J. A. O'Sullivan, "Information-Theoretic Analysis of Watermarking," *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'00)*, Istanbul, Turkey, June 2000.

P. Ishwar and P. Moulin, "Fundamental Equivalences between Set-Theoretic and Maximum-Entropy methods in Multiple-Domain Image Restoration," *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'00)*, Istanbul, Turkey, June 2000.

J. C. Ye, Y. Bresler and P. Moulin, "Global Confidence Regions in Parametric Shape Estimation," *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'00)*, Istanbul, Turkey, June 2000.

A. Jain, P. Moulin, M. I. Miller and K. Ramchandran, "Information-Theoretic Bounds on Target Recognition Performance," *SPIE's 14th International Symposium on AeroSense 2000*, Orlando, FL, April 2000.

J. C. Ye, Y. Bresler and P. Moulin, "Wavelet Descriptors for Parametric Shape Estimation in Nonlinear Inverse Scattering Problems," *SPIE's International Symposium on AeroSense 2000*, Orlando, FL, April 2000.

P. Ishwar and P. Moulin, "Multiple-Domain Image Modeling and Restoration," *Proc. IEEE Int. Conf. on Image Proc.*, Vol. I, pp. 362—366, Kobe, Japan, 1999.

J. Liu and P. Moulin, "Image Denoising Based on Scale-Space Mixture Modeling of Wavelet Coefficients," *Proc. IEEE Int. Conf. on Image Proc.*, Vol. I, pp. 386—390, Kobe, Japan, 1999.

S. Jana and P. Moulin, "Optimal Design of Transform Coders for Image Classification," *Proc. Conf. on Info. Syst. and Sci.*, Baltimore, MD, March 1999.

A. Jain, P. Moulin, K. Ramchandran and M. I. Miller, "Performance Bounds for ATR based on Compressed Data," *Proc. Army Compression Workshop*, Huntsville, AL, Dec. 1998.

P. Moulin, M. Anitescu, and K. Ramchandran, "Some Fundamental Properties of MMSE Filter Banks," *Proc. IEEE Sig. Proc. Symp. on Time-Frequency and Time-Scale Analysis*, Pittsburgh, PA, Oct. 1998.

P. Moulin and J. Liu, "Analysis of Multiresolution Image Denoising Schemes Using Generalized - Gaussian Priors," *Proc. IEEE Sig. Proc. Symp. on Time-Frequency and Time-Scale Analysis*, Pittsburgh, PA, Oct. 1998.

J. Liu and P. Moulin, "A New Complexity Prior for Multi-Resolution Image Denoising," *Proc. IEEE Sig. Proc. Symp. on Time-Frequency and Time-Scale Analysis*, Pittsburgh, PA, Oct. 1998.

J. Liu and P. Moulin, "Complexity-Regularized Image Restoration," *Proc. IEEE Int. Conf. on Image Proc. (ICIP'98)*, Chicago, IL, Oct. 1998.

M. K. Mihçak, K. Ramchandran and P. Moulin, "Adaptive Wavelet Packet Image Coding Using an Estimation-Quantization Framework," *Proc. IEEE Int. Conf. on Image Proc. (ICIP'98)*, Vol. I, pp. 92—96, Chicago, IL, Oct. 1998.

P. Moulin and M. Miller, "Detection Bounds for Automatic Target Recognition," *Proc. Int. Symp. on Information Theory (ISIT'98)*, Boston, MA, Aug. 1998.

J. A. O'Sullivan, P. Moulin and M. Ettinger, "Information Theoretic Analysis of Steganography," *Proc. Int. Symp. on Information Theory (ISIT'98)*, Boston, MA, Aug. 1998.

X. Yang, K. Ramchandran and P. Moulin, "Optimization of High-Energy-Compaction, Nearly-Orthonormal, Linear-Phase Filter Banks," *Proc. ISCAS'98*, Monterrey, CA, June 1998.

P. Ishwar, K. Ratakonda, P. Moulin and N. Ahuja, "Image Denoising in Multiple Compaction Domains," invited paper, *Proc. ICASSP'98*, Vol. 3, pp. 1889—1892, Seattle, May 1998.

P. Moulin, M. Anitescu and K. Ramchandran, "Asymptotic Performance of Subband Coders Using Constrained, Signal-Adapted FIR Filter Banks," *Proc. 32nd Conf. on Info. Syst. and Sci. (CISS)*, Princeton, NJ, March 1998.

J. Liu and P. Moulin, "Complexity-Regularized Image Denoising," *Proc. IEEE Int. Conf. on Image Proc. (ICIP'97)*, pp. II. 370—373, Santa Barbara, CA, Oct. 1997.

P. Ishwar and P. Moulin, "Switched Control-Grid Interpolation for Motion-Compensated Video Coding," *Proc. IEEE Int. Conf. on Image Proc. (ICIP'97)*, pp. III. 650—653, Santa Barbara, CA, Oct. 1997.

R. Krishnamurthy, J. W. Woods and P. Moulin, "Compactly-Encoded Optical Flow Fields for Bidirectional Prediction and Video Coding," *Proc. VCIP'97*, Feb. 1997.

P. Moulin, K. Ramchandran and V. Pavlovic, "Transform Image Coding based on Joint Adaptation of Filter Banks and Tree Structures," *Proc. IEEE Int. Conf. on Image Proc. (ICIP'96)*, Lausanne, Switzerland, Sep. 1996.

R. Krishnamurthy, P. Moulin and J. W. Woods, "Multiscale Motion Estimation for Scalable Video Coding," *Proc. ICIP'96*, Lausanne, Switzerland, pp. I. 965—968, Sep. 1996.

P. Moulin, "Signal Estimation Using Adapted Tree-Structured Bases and the MDL Principle," *Proc. IEEE Sig. Proc. Symp. on Time-Frequency and Time-Scale Analysis*, pp. 141—143, Paris, June 1996.

P. Moulin, "Design of Signal-Adapted FIR Paraunitary Filter Banks," *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'96)*, Atlanta, GA, May 1996.

P. Moulin and R. Krishnamurthy, "Multiscale Motion Fields for Video Coding," *Proc. 9th IEEE Workshop on Image and Multidim. Signal Processing*, Belize, March 1996.

R. Krishnamurthy, P. Moulin and J. Woods, "Optical Flow Techniques Applied to Video Coding," *Proc. ICIP'95*, Washington, D.C., Oct. 1995.

P. Moulin, "Model Selection Criteria and the Orthogonal Series Method for Function Estimation," *Proc. IEEE Int. Symp. on Information Theory (ISIT'95)*, Whistler, B.C., Sep. 1995.

P. Moulin, "A New Look at the Signal-Adaptive QMF Bank Design," *Proc. ICASSP'95*, Detroit, MI, pp. 1312–1315, Apr. 1995.

P. Moulin, "A Relaxation Technique for Minimizing the L^2 Reconstruction Error in Nonorthogonal Subband Coding," *Proc. ICIP'94*, Austin, TX, Nov. 1994.

P. Moulin, "The Use of Wavelets for Spectral Density Estimation with Local Bandwidth Adaptation," *Proc. ISIT'94*, p. 40, Trondheim, Norway, June 1994. Also invited paper, *Interface'94*, Research Triangle Park, June 1994.

P. Moulin, "Optimization Techniques for Efficient Image Compression in Nonorthogonal Multiresolution Bases," *Proc. 1994 NJIT Symp. on Wavelets*, Apr. 1994.

P. Moulin, "Wavelets as a Regularization Technique for Spectral Density Estimation," *Proc. SIAM Annual Meeting*, p. 69, Philadelphia, July 1993.

P. Moulin, I. Rhee and P. Caputo, "Video Signal Processing on SIMD Parallel Computer Architectures," *Proc. 8th IEEE Workshop on Image and Multidim. Signal Processing*, pp. 246–247, Cannes, France, Sep. 1993.

P. Moulin and A. Loui, "Hierarchical-Finite-Element Estimation and Parameterization of Optical Flow Field," *Proc. 8th IEEE Workshop on Image and Multidim. Signal Processing*, pp. 130–131, Cannes, France, Sep. 1993.

P. Moulin and A. Loui: "Application of a Multiresolution Optical Flow Based Method for Motion Estimation to Video Coding," *Proc. ISCAS'93*, pp. I.1–4, Chicago, IL, May 1993.

P. Moulin, "Optimal L^2 Approximation of Images in Nonorthogonal Multiresolution Bases," *Proc. ICASSP'93*, pp. V.189–192, Minneapolis, MN, Apr. 1993.

P. Moulin, "Model-Based Multiresolution Restoration of Speckle Images: Application to Radar Imaging," invited paper, *Proc. IEEE Int. Symp. on Information Theory*, p. 67, San Antonio, TX, Jan. 1993.

P. Moulin, "Adaptive Multiresolution Image Restoration and Compression," invited paper, *IEEE EMBS Workshop on Case Studies in Medical Instrument Design*, Orlando, FA, Oct. 91.

P. Moulin, "Wavelets as a Regularization Technique for Spectral Density Estimation," *Proc. IEEE Signal Processing Conf. on Time-Frequency and Time-Scale Analysis*, pp. 73–76, Victoria, B.C., Oct. 1992.

P. Moulin, "An Adaptive Finite-Element Method for Image Representation," *Proc. Int. Conf. on Pattern Recognition*, pp. III.70–74, The Hague, The Netherlands, Aug. 1992.

P. Moulin, "An Adaptive Spline Method for Signal Restoration," *Proc. ICASSP'92*, San Francisco, CA, Mar. 92, pp. IV.169–172.

P. Moulin, J. A. O'Sullivan, and D. L. Snyder, "A Method of Sieves for Estimating the Spectral Density," *Proc. IEEE Int. Symp. on Information Theory*, p. 67, Budapest, Hungary, June 91.

J. A. O'Sullivan, P. Moulin, D. L. Snyder, and S. P. Jacobs: "Computational Considerations for Maximum-Likelihood Radar Imaging," *Proc. 24th CISS Conf.*, Princeton, NJ, March 1990.

P. Moulin, D. L. Snyder, and J. A. O'Sullivan: "A Method of Sieves for Regularizing Maximum-Likelihood Spectral Estimates," *Proc. IEEE Int. Symp. on Information The-*

ory, p. 112, San Diego, CA, Jan. 1990.

P. Moulin, D. L. Snyder, and J. A. O'Sullivan: "A Sieve-Constrained Maximum-Likelihood Estimator for the Spectrum of a Gaussian Process," *Proc. 28th Allerton Conf.*, Urbana-Champaign, IL, Sept. 1989.

J. A. O'Sullivan, D. L. Snyder, and P. Moulin, "The Role of Spectrum Estimation in Forming High-Resolution Radar Images," *Proc. ICASSP 1989*, Glasgow, U.K., May 1989.

P. Moulin, J. A. O'Sullivan, and D. L. Snyder, "Maximum-Likelihood Spectrum Estimation of Periodic Processes from Noisy Data," *Proc. CISS Conf.*, Johns Hopkins University, Baltimore, MD, March 1989.

J. A. O'Sullivan, P. Moulin, and D. L. Snyder, "Cramer-Rao Bounds for Constrained Spectrum Estimation with Application to a Problem in Radar Imaging," *Proc. 27th Allerton Conf.*, Urbana-Champaign, IL, Sept. 1988.

R. Boite and P. Moulin, "Comparaison du rapport signal à bruit pour une représentation numérique en virgule fixe et en virgule flottante," *Proc. IASTED Symp.*, Paris, France, June 1985.

EXHIBIT B

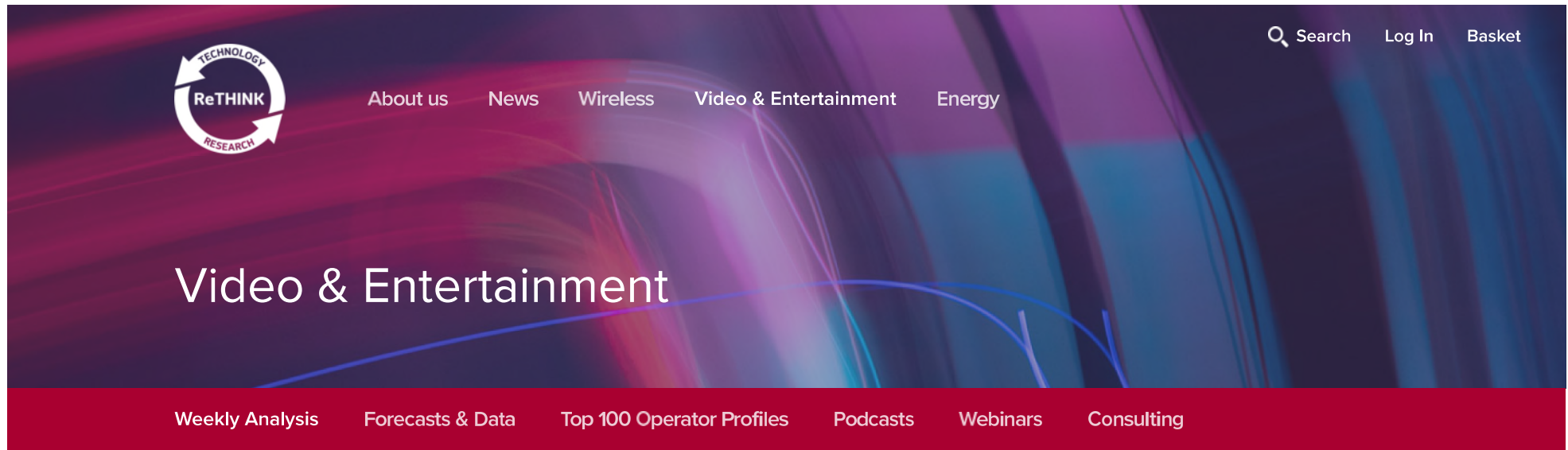
Materials Considered

[1] Ping Hu, Weiqiang Wang, and Ke Lu, “Video Text Detection with Text Edges and Convolutional Neural Network, 3rd IAPR Asian Conference on Pattern Recognition (ACPR),” November 2015, available at <https://ieeexplore.ieee.org/document/7486588>.

[2] Cong Yao, Jianan Wu, Xinyu Zhou, Chi Zhang, Shuchang Zhou, Zhimin Cao, Qi Yin, “Incidental Scene Text Understanding: Recent Progresses on ICDAR 2015 Robust Reading Competition Challenge 4,” February 2016, available at <https://arxiv.org/pdf/1511.09207.pdf>.

[3] Adam Slucki, Tomasz Trzcinski, Adam Bielski., and Pawel Cyrta, “Extracting textual overlays from social media videos using neural networks,” April 2018, available at <https://arxiv.org/pdf/1804.10687v2.pdf>.

EXHIBIT C



25 March 2021

Hyphametrics comes out swinging with hyper-surveillance attribution



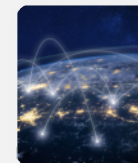
By **Rafi Cohen**

Hyphametrics is an audience measurement start-up that does not take the buzzwords 'cross-platform' and 'holistic' lightly. While still in its infancy, Hyphametrics could assemble a highly accurate dataset that will be hard to match in terms of transparency, if all goes to plan.

This prediction is based on how Hyphametrics performs hyper-surveillance on every aspect of digital entertainment in the home by scrupulously tracking everything from the feed displayed on the TV, to the IP traffic on the router, and even the locations of people in the house. It sounds spooky, but this Big Brother-esque measurement is reserved for a select set of opt-in panel audiences.

Talking to Faultline this week, Hyphametrics CEO, Joanna Drews, and CTO, Gerardo

Forecasts & Data
from **Rethink TV**



6 October 2022

**Video Delivery Market
Forecast 2022-2028**



25 August 2022

Operator Profiles Database



1 August 2022

**Open Internet Contribution
Transport Protocol Forecast**

Lopez, talked us through the ins and outs of its proprietary system. At the hub of the house is the coreMeter box, which hosts multiple HDMI inputs for outgoing feeds from all TVs, gaming consoles, and set tops. Frame by frame analysis is conducted on those feeds, with different techniques applied depending on the source of the media.

A set top feed will be examined for broadcaster logos, as well as the channel information overlays that appear when a channel is changed, whereas a games console feed will be analyzed for browsing of the console interface to identify which game is selected, as well as developer logos at the start of gameplay.

Similarly, user choices on OTT platforms are tracked by following the user path towards content. Both Drews and Lopez feel this makes Hyphametrics the only measurement vendor that truly knows what people are watching. “We know that it is Season 3, Episode 6 of Seinfeld, not just a random sitcom,” Lopez told us.

Keen to poke some holes, we asked whether Hyphametrics would be able to detect a pirated streaming website that is being displayed on the TV by either casting or HDMI? Lopez told us that while coreMeter would not be able detect this through its HDMI inputs, the wider system would be able to do so through IP traffic tracking.

The coreMeter is multi-tasking several capabilities simultaneously. As well as being an interface for HDMI feeds, the box acts as a router for network traffic measurement, and a quasi-connected TV from which OTT applications can be run (and measured.)

“We are replicating the CTV experience from our box. Any app they can have in their smart TV they can download here,” Drews explained.

‘Source of truth’ is another buzzterm that often makes us switch off when hearing from measurement vendors, but Drews was keen to prove that Hyphametrics was not just spouting marketing fluff.

“We are never making a guess or an inference. While many competitors are reliant on libraries and schedule data, so there is always an inference made in the methodology, we are officially measuring who is in the room,” Drews assured us.

Take measurement rival Samba TV, which Drews felt was a fitting example. “Samba TV works through TV manufacturers, which allows for scale, but it blocks measurement on all the top apps, and cannot tell you how many people are watching,” she explained. “It makes ID matches with partners like Xperi, but those are loaded with inferences. It is not enough to provide granularity.”

Hyphametrics can also tell who in the house is watching the TV at any moment by tracking the presence of users’ mobile phones via Bluetooth and WiFi, which panelists

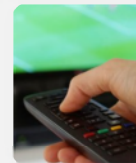


Transport Protocol Forecast 2022-2027



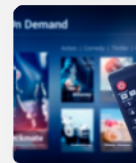
29 June 2022

Content Recommendation Engines Forecast 2022- 2027



25 May 2022

Low Latency Distribution 2022-2027



8 April 2022

SVoD & AVoD Forecast 2022-2027



31 January 2022

Multicast-ABR Market Forecast 2022-2027



7 December 2021

WiFi Cloud Management, Multi-AP, WiFi 6, 6E & 7 Forecast— 2021-2026

agree to have turned on at all times. “We are officially measuring who is in the room and what they are exposed to,” Lopez assured us.

Naturally, any company claiming such a competitive edge should be appropriately protected. Hyphametrics has just been issued its first patent, which Drews said covers 15 unique elements of its methodology, including the tracking of IP traffic, the use of certain ML algorithms, as well as the proprietary hardware, software and cloud technologies that are used.

Hyphametrics is in the process of recruiting its first preliminary panel of 100 households. “We know the market expectation in terms of neutrality, and we do not augment households. There will be some with every SVoD app, and other with just an antenna,” explained Drews, meaning that a panel household is left as is, with no extra SVoD subscriptions, games consoles or set tops added.

Drews expects the panel to be reflective of the US within the first few hundred households, and once the panel hits 5,000 households, clients will be able to integrate its findings with census data to produce ratings.

So, who are these clients? Drews was unable to name any of the data platforms and media companies that were showing interest, but did say that its main focus is streaming platforms. “Networks are pumping ad dollars into those spaces, so those platforms need to better understand the user and their competitors,” she explained. “What is a Pluto TV user doing when not on the platform?”

Interestingly, Drews did say that the likes of Nielsen and Comscore would find value in Hyphametrics’ offering, suggesting it is not rival but rather a complementary service.

Being early days, Hyphametrics is not breaking out data on certain trends – like Nielsen and Comscore have done with D2C movies – but this is coming. “We have already entered into market conversations with interest in specific subsets. We have a proprietary system built from scratch, so it is a small lift,” Drews told us.

Early interest is in IP traffic measurement, specifically seeing how effective ad placement matches internet browsing, as well as the more general interplay between all entertainment devices in the home.

Like what you're reading?

[Request a Faultline Trial](#)

Bitmovin weighs in on guiding Eventive’s hypergrowth

Attempts by NBCU, Roku to disrupt ad formats fall flat



To contact Rethink please email us at info@rethinkresearch.biz
Alternatively call us on +44 (0)1179 257019



EXHIBIT D

NEXT TV

Broadcasting + Cable
Multichannel News

Broadcasting+Cable




TRENDING

New 'Hot Bench' Judges

The Death of Linear CPMs?

Holiday Programming

GAC App Debuts

[Home](#) > [News](#)

HyphaMetrics Issued Patent for Cross-Platform Measurement

By [Jon Lafayette](#) ([Broadcasting & Cable](#)) published March 10, 2021

Single-source system based on optical recognition, A.I. and machine learning

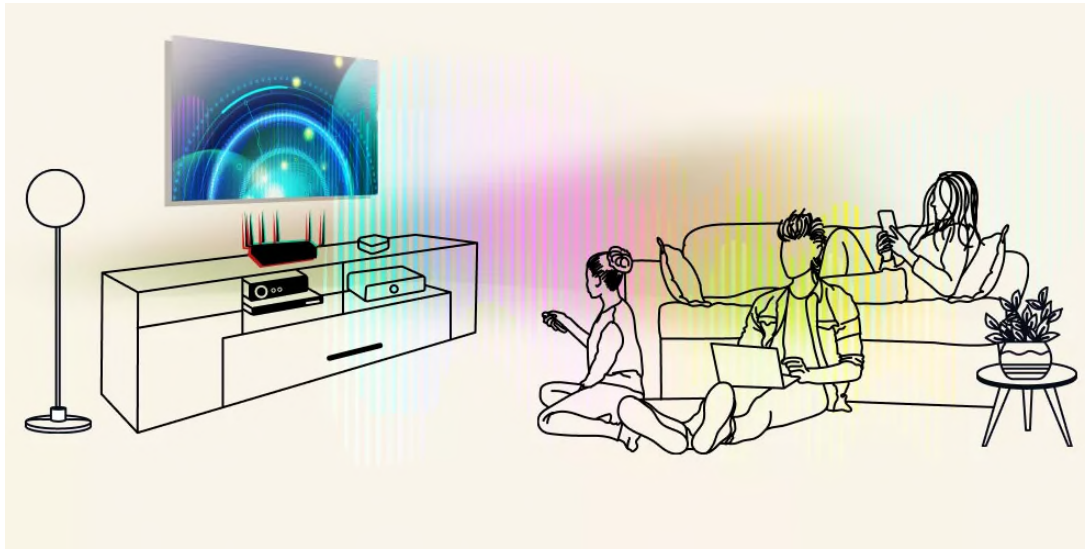


BROADCASTING & CABLE
NEWSLETTER

- ☐ I consent the use of my personal data in accordance with Future's [privacy policy](#), including its cookie policy.

SIGN ME UP

By submitting your information you agree to the [Terms & Conditions](#) and [Privacy Policy](#) and are aged 16 or over.



HyphaMetrics got a patent for its cross-platform measurement system (Image credit: HyphaMetrics)

HyphaMetrics, an independent media measurement company, was issued a U.S. patent for [its next-generation cross-platform measurement system](#).


Patent No. 10,932,002 covers the company's unique coreMeter hardware, its methodology for collecting data from all media sources in a household, and how the company determines individuals' media consumption within a household.

HyphaMetrics' system "can facilitate the measurement of everything occurring in someone's home. We're able to measure across all the walled gardens, across all the various silos," said CEO and co-founder Joanna Drews. "What that produces is a true cross-platform single-source metric."

The patent will enable HyphaMedia to deploy a "complete proprietary measurement system



Advertisement



**You're invested
in Astoria.
So are we.**

Say hello to your new
neighborhood bank.

GET TO KNOW US

Advertisement

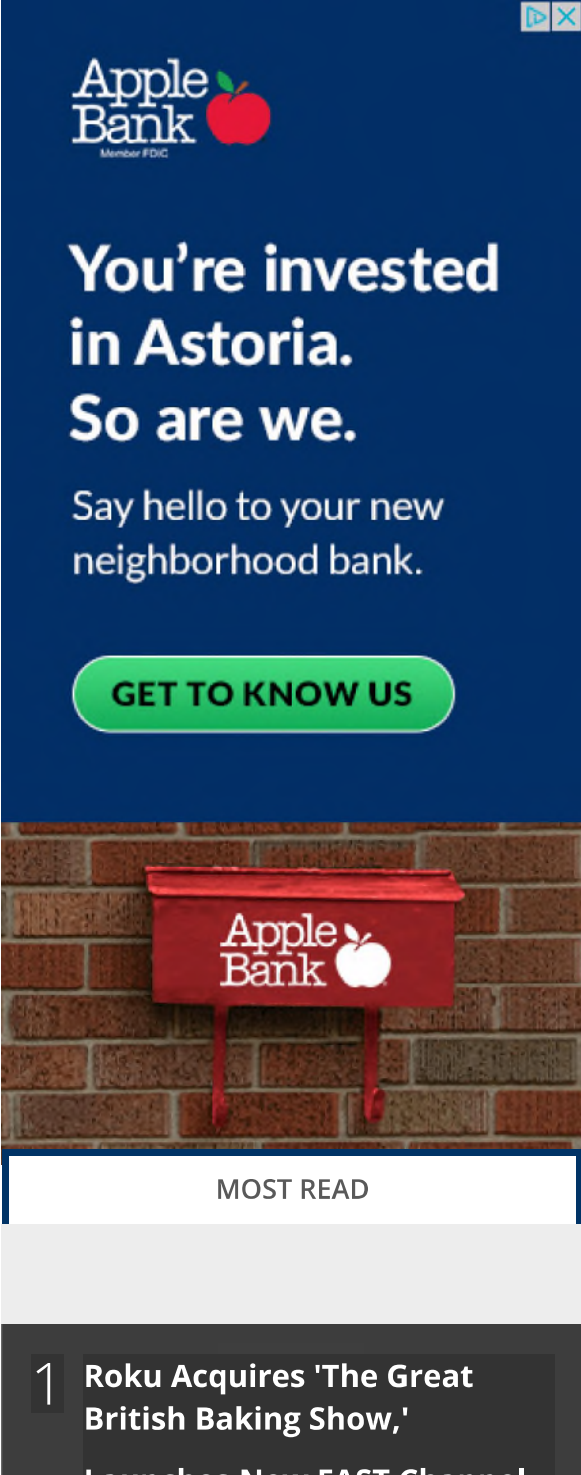


that is absolutely going to solve for a lot of the Holy Grail issues about omni-channel, cross-screen person-level measurement that the entire industry has been debating and asking for for a very long time," said Mike Bologna, the long-time advanced advertising head at GroupM who joined HyphaMetrics as chief revenue officer last year.

Joanna Drews, co-founder and CEO, HyphaMetrics
(Image credit: HyphaMetrics)

Drews was director of product management and emerging assets at Comscore and a partner in GroupM's research practice.

Large measurement companies including Nielsen and Comscore are working on their own cross-media measurement systems.



The advertisement for Apple Bank features a dark blue banner at the top with the Apple Bank logo (a red apple with a bite taken out of it) and the text "Member FDIC". Below the banner, the text reads: "You're invested in Astoria. So are we." followed by "Say hello to your new neighborhood bank." and a green button that says "GET TO KNOW US". Below the banner is a red mailbox mounted on a brick wall, also featuring the Apple Bank logo. At the bottom of the advertisement, there is a section titled "MOST READ" with a list of articles. The first article is "1 Roku Acquires 'The Great British Baking Show,' Launches New FAST Channel".

Apple Bank
Member FDIC

**You're invested
in Astoria.
So are we.**

Say hello to your new
neighborhood bank.

GET TO KNOW US

MOST READ

1 Roku Acquires 'The Great British Baking Show,' Launches New FAST Channel

Drews sees HyphaMetrics as a compliment for those systems, rather than a replacement.

"All of those solutions are really great at scale. They lack the granularity and the definition that we provide," she said, adding someday, Nielsen might be one of HyphaMetric's customers.

HyphaMetrics is in pilot mode now, according to Bologna.



Michael Bologna
(Image credit:
HyphaMetrics)

"We are about to launch a field trial," he said, putting its coreMeter boxes into about 100 homes while working with a few select partners." He did not name the partners.

Bologna noted that the boxes are costly and that it's expensive to recruit panels. The company previously raised \$2 million, but will be raising

additional funds via an equity round of financing in order to continue growing.

A lot of measurement systems use audio content recognition or automatic content recognition to determine what's being watched. HyphaMetrics' technology uses artificial intelligence and machine learning to read what's on the screen, Drews said. Its boxes are attached to each TV in a household. The boxes also have a router that enabled them to determine what's being watched on streaming devices.

The systems allow HyphaMetrics to determine co-viewing, viewing personas and household viewing, Drews said.

HyphaMetrics' three products so far are ClearviewMetrics, which measures unduplicated viewing data at the individual level across every network, program, advertisement, product placement, streaming app, gaming environment for every single device in the household; ContentMetrics,

Launches New FAST Channel Around It

2 NBCU Says Clients Using iSpot as Currency for Upfront Deals

3 Paramount Plus Claims Its Biggest Signup Day Ever on NFL Football and the 'Taylor Sheridan Universe'

4 'Baking It' Season Two Starts With Holiday Special

5 Showtime Sets January Premiere Date for 'Your Honor' Second Season

Advertisement

which collects real time data about who is watching what content and in what format -- live, time-shifted, or streaming -- on TVs within an entire household; and MobileMetrics, granular data which demonstrates time spent in apps, simultaneous app usage, multitasking metrics, and more across mobile media environments.



Jon Lafayette  

Jon has been business editor of Broadcasting+Cable since 2010. He focuses on revenue-generating activities, including advertising and distribution, as well as executive intrigue and merger and acquisition activity. Just about any story is fair game, if a dollar sign can make its way into the article. Before B+C, Jon covered the industry for TVWeek, Cable World, Electronic Media, Advertising Age and The New York Post. A native New Yorker, Jon is hiding in plain sight in the suburbs of Chicago.

MORE ABOUT BROADCASTING AND CABLE



**Football,
'Yellowstone' and
Election Coverage:**

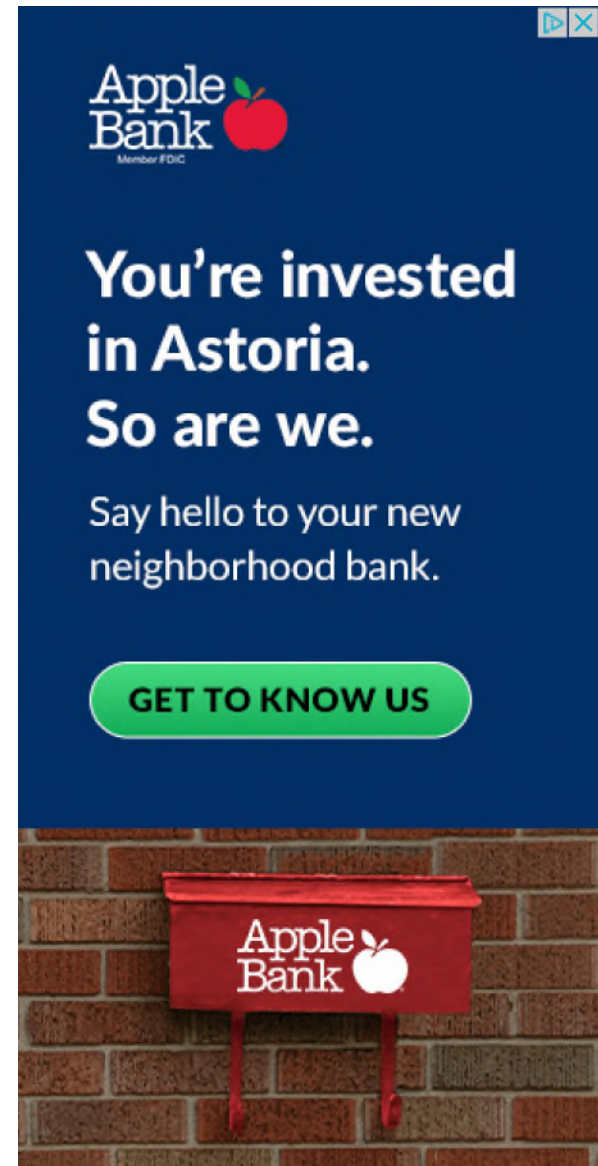


**NBCU Says Clients
Using iSpot as**

LATEST



**Paramount
Network Again
Gives**



Apple Bank
Member FDIC

**You're invested
in Astoria.
So are we.**

Say hello to your new
neighborhood bank.

GET TO KNOW US

TV By the Numbers ►

Currency for Upfront Deals ►

'Yellowstone' TV's Biggest Promo Push ►

SEE MORE LATEST ►

MOST POPULAR



HBO Greenlights Third Season of 'The White Lotus'

By [R. Thomas Umstead](#) about 2 hours ago



Fox Sports's World Cup Coverage Plays on Broadcast and Cable Field

By [R. Thomas Umstead](#) about 3 hours ago



Faith Jessie Joins WXIA-TV In Atlanta As Weeknight Anchor

By [Jon Lafayette](#) about 3 hours ago



NewsNation Aims Billboard at CNN Staffers Facing Layoffs

By [Jon Lafayette](#) about 3 hours ago

FCC Unveils New Broadband Map


By [John Eggerton](#) about 5 hours ago

Bethenny Frankel Boards Kevin O'Leary's 'Money Court'

By [Michael Malone](#) about 5 hours ago

Hallmark Channel Sets January Premiere Date for 'The Way Home'

By [R. Thomas Umstead](#) about 15 hours ago




YouTube Leads All OTT Platforms, Including Netflix, in Monthly Average Viewers (Ridiculous Chart of the Day)

By [Daniel Frankel](#) about 19 hours ago



FCC Approves Broadband Label Rules

By [John Eggerton](#) about 21 hours ago



'America's Got Talent: All-Stars' Reveals Acts That Will Compete

By [Michael Malone](#) about 22 hours ago

Broadcasting+Cable is part of Future US Inc, an international media group and leading digital publisher. **Visit our corporate site.**

[Terms and conditions](#)

[Privacy policy](#)

[Cookies policy](#)

[Accessibility statement](#)

[Careers](#)

[Advertise with Us](#)

[Contact Us](#)

[Do not sell my info](#)

© Future US, Inc. Full 7th Floor, 130 West 42nd Street, New York, NY 10036.

EXHIBIT E



US010932002B2

(12) **United States Patent**
Zamudio et al.

(10) **Patent No.:** **US 10,932,002 B2**

(45) **Date of Patent:** **Feb. 23, 2021**

(54) **CROSS-MEDIA MEASUREMENT DEVICE AND METHOD**

21/4402 (2013.01); **H04N 21/4662** (2013.01);
H04N 21/812 (2013.01); **G06K 2209/25**
(2013.01); **G06Q 30/0242** (2013.01); **G06Q**
30/0272 (2013.01)

(71) Applicant: **Hyphametrics, Inc.**, Albany, NY (US)

(72) Inventors: **Gerardo Lopez Zamudio**, Mexico City (MX); **Joanna Drews**, Derwood, MD (US)

(73) Assignee: **Hyphametrics, Inc.**, Albany, NY (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **16/925,029**

(22) Filed: **Jul. 9, 2020**

(65) **Prior Publication Data**

US 2021/0014565 A1 Jan. 14, 2021

Related U.S. Application Data

(60) Provisional application No. 62/871,789, filed on Jul. 9, 2019.

(51) **Int. Cl.**

H04H 60/32 (2008.01)

H04N 21/442 (2011.01)

G06K 9/00 (2006.01)

H04N 21/258 (2011.01)

H04N 21/4402 (2011.01)

H04N 21/466 (2011.01)

G06N 20/00 (2019.01)

G06N 5/04 (2006.01)

H04N 21/81 (2011.01)

G06Q 30/02 (2012.01)

(52) **U.S. Cl.**

CPC ... **H04N 21/44222** (2013.01); **G06K 9/00744** (2013.01); **G06N 5/04** (2013.01); **G06N 20/00** (2019.01); **H04N 21/25883** (2013.01); **H04N**

(58) **Field of Classification Search**

CPC **H04N 21/44222**; **H04N 21/25883**; **H04N**
21/4402; **H04N 21/4662**; **H04N 21/812**

USPC **725/14**

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

9,055,309 B2 * 6/2015 Neumeier **H04N 21/23614**
9,705,544 B2 * 7/2017 van de Beek **H04L 27/06**
10,375,451 B2 * 8/2019 Neumeier **H04N 21/25866**
10,631,068 B2 * 4/2020 Navin **G08C 17/02**

(Continued)

Primary Examiner — Pankaj Kumar

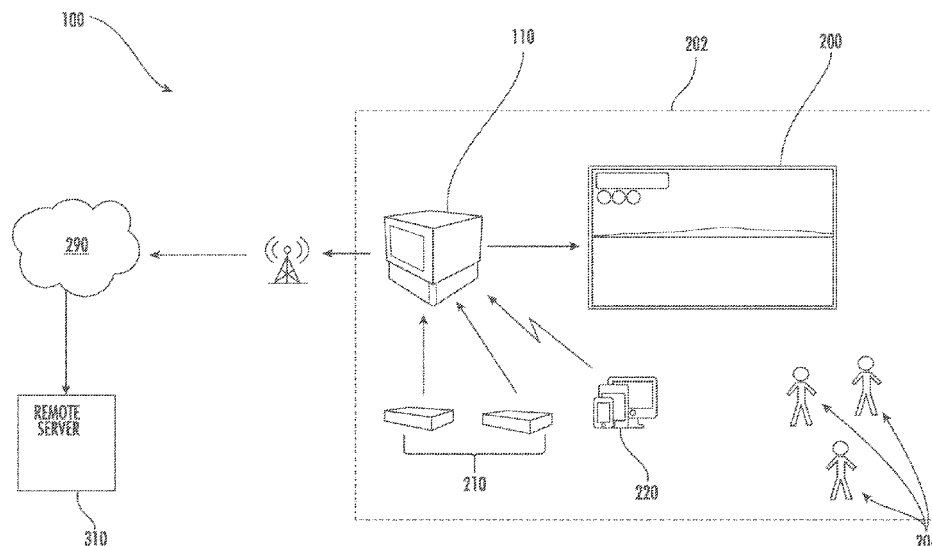
Assistant Examiner — Sahar Aqil Riaz

(74) *Attorney, Agent, or Firm* — Maginot, Moore & Beck LLP

(57) **ABSTRACT**

A method of identifying media content presented on a display device includes determining a selected input source providing a video signal to the display device, and then selecting a first set of content identification rules when it is determined that the selected input source is a first input source, and selecting a second set of content identification rules when it is determined that the selected input source is a second input source. The method further comprises applying the selected first set or second set of content identification rules to the video signal in order to generate content identification data for the media content presented on the display device. Application of the content identification rules includes waiting for a trigger event and applying an algorithm to one or more frames of the video signal following the trigger event.

20 Claims, 43 Drawing Sheets



US 10,932,002 B2

Page 2

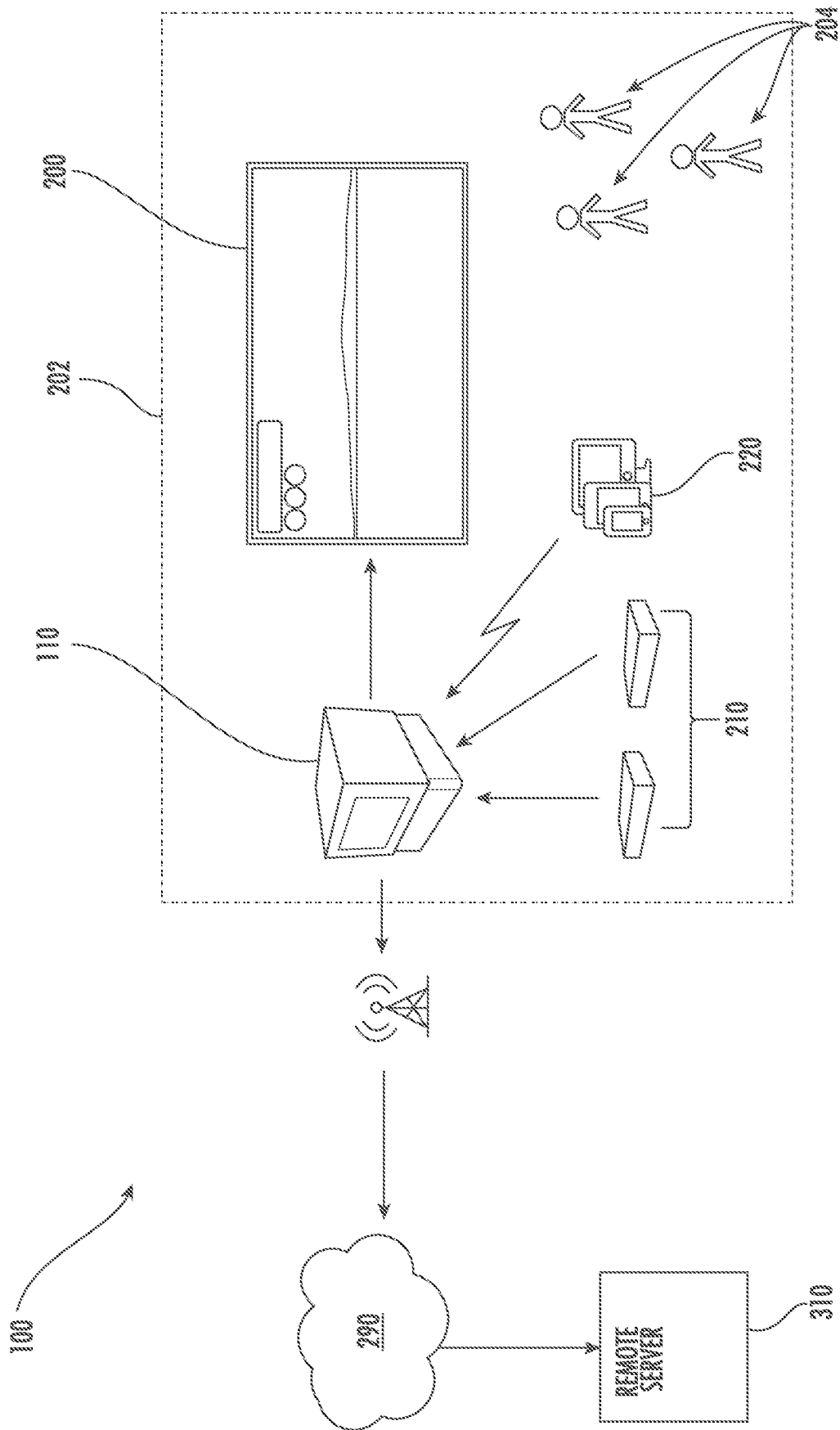
(56)

References Cited

U.S. PATENT DOCUMENTS

10,779,058	B2 *	9/2020	Panchaksharaiah	H04N 21/44222
2008/0184245	A1 *	7/2008	St-Jean	G06K 9/00771
				718/103
2011/0078719	A1 *	3/2011	Kenyon	G06K 9/0055
				725/19
2012/0158726	A1 *	6/2012	Musgrove	G06F 16/353
				707/737
2014/0225924	A1 *	8/2014	Loxam	G06T 19/006
				345/633
2014/0317155	A1 *	10/2014	Treibach-Heck	G06F 16/95
				707/811
2018/0316939	A1 *	11/2018	Todd	H04N 21/6587
2018/0316947	A1 *	11/2018	Todd	H04N 21/25875
2019/0195652	A1 *	6/2019	Lafon	G08G 5/0021
2020/0162788	A1 *	5/2020	Cremer	H04N 21/442
2020/0228880	A1 *	7/2020	Iyer	H04N 21/23424
2020/0236415	A1 *	7/2020	Khanna	H04N 21/2353
2020/0296472	A1 *	9/2020	Chung	H04N 21/4826

* cited by examiner



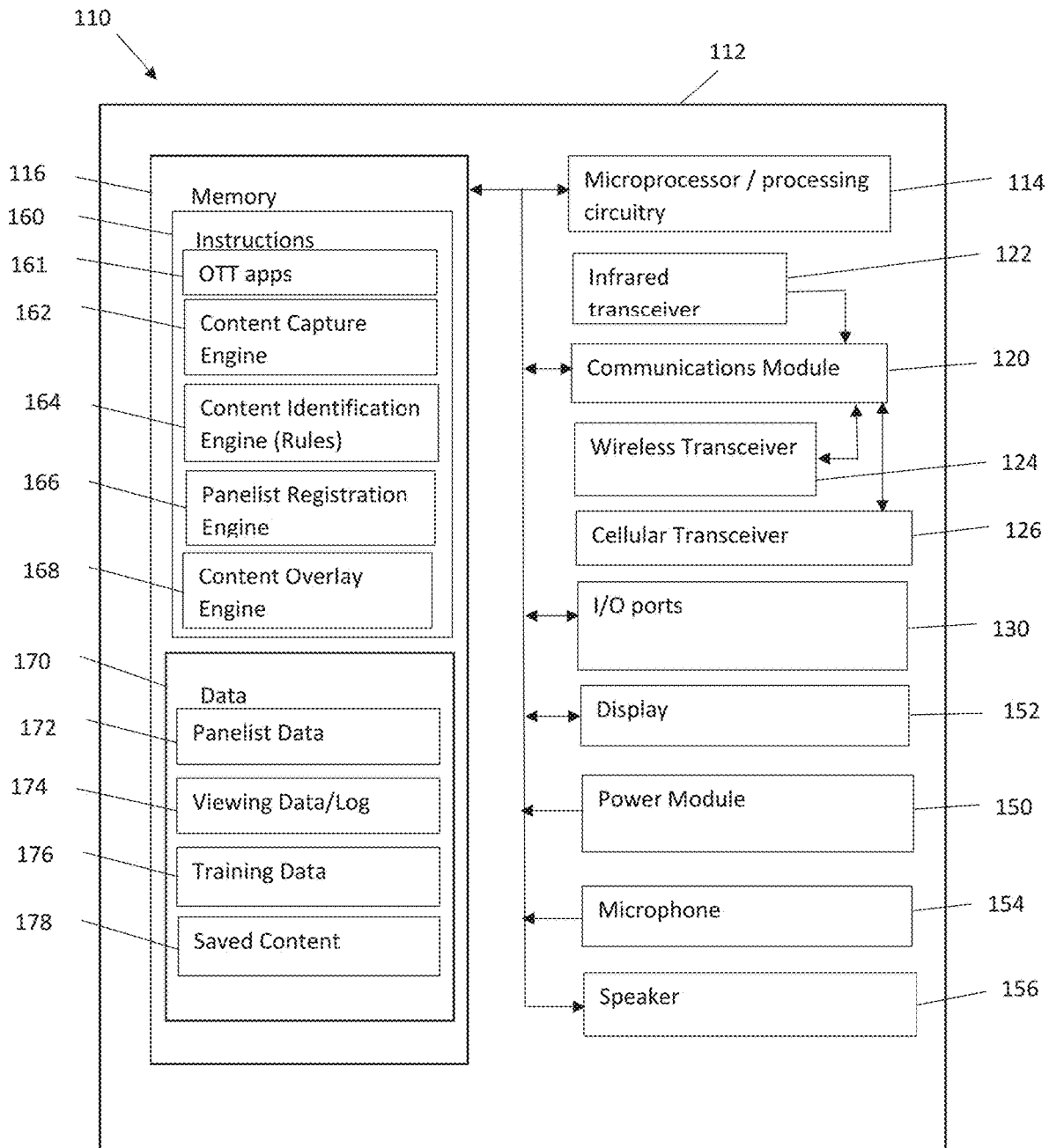
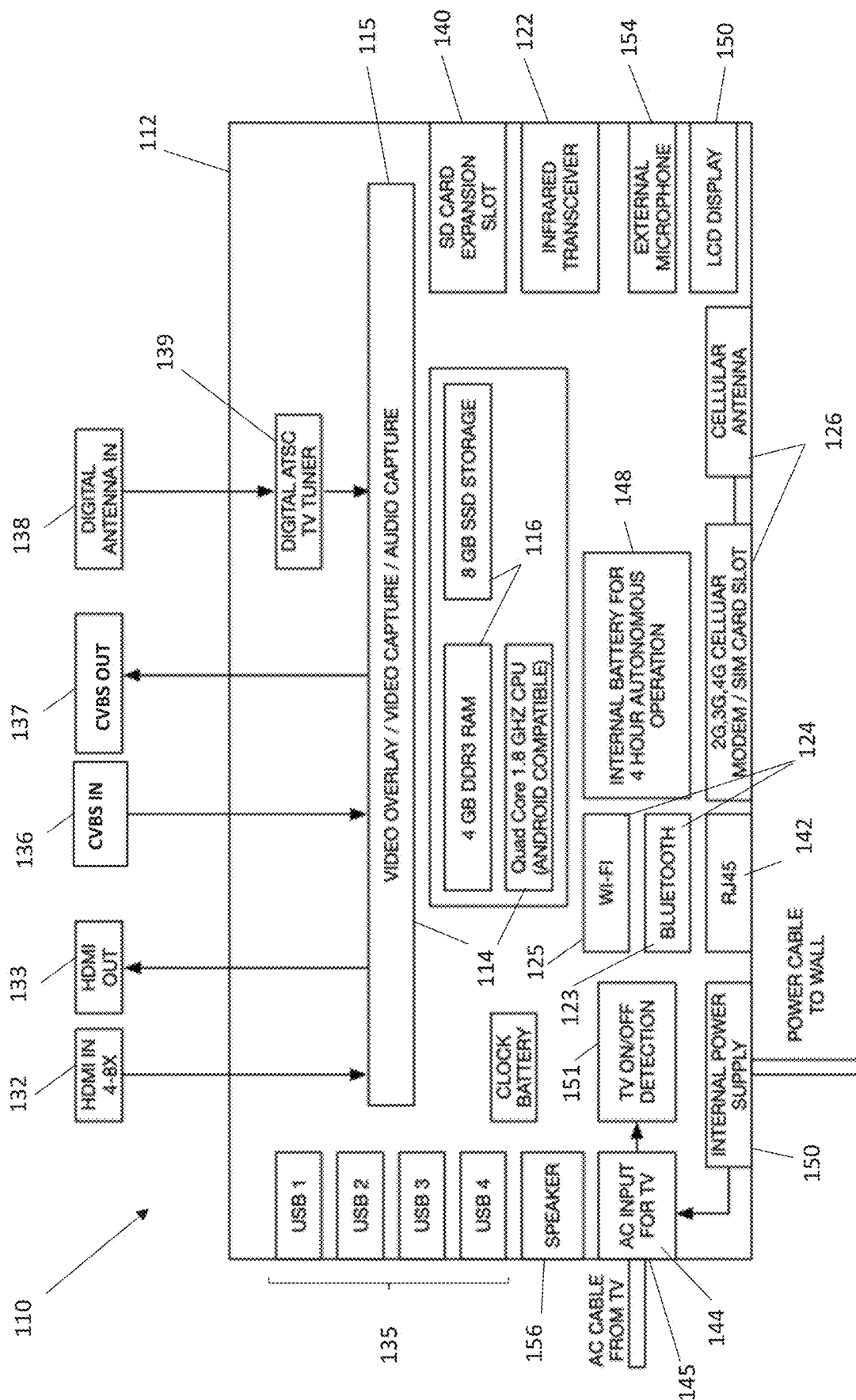


FIG. 2A



W. G. 2B

U.S. Patent

Feb. 23, 2021

Sheet 4 of 43

US 10,932,002 B2

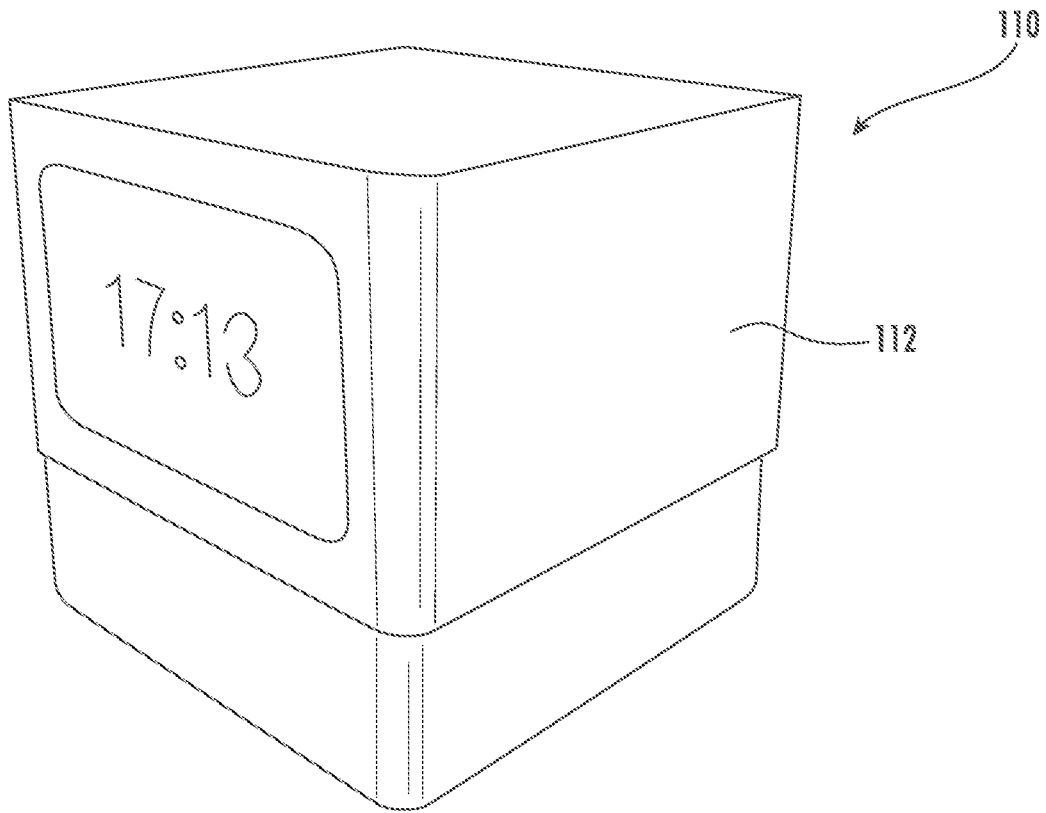


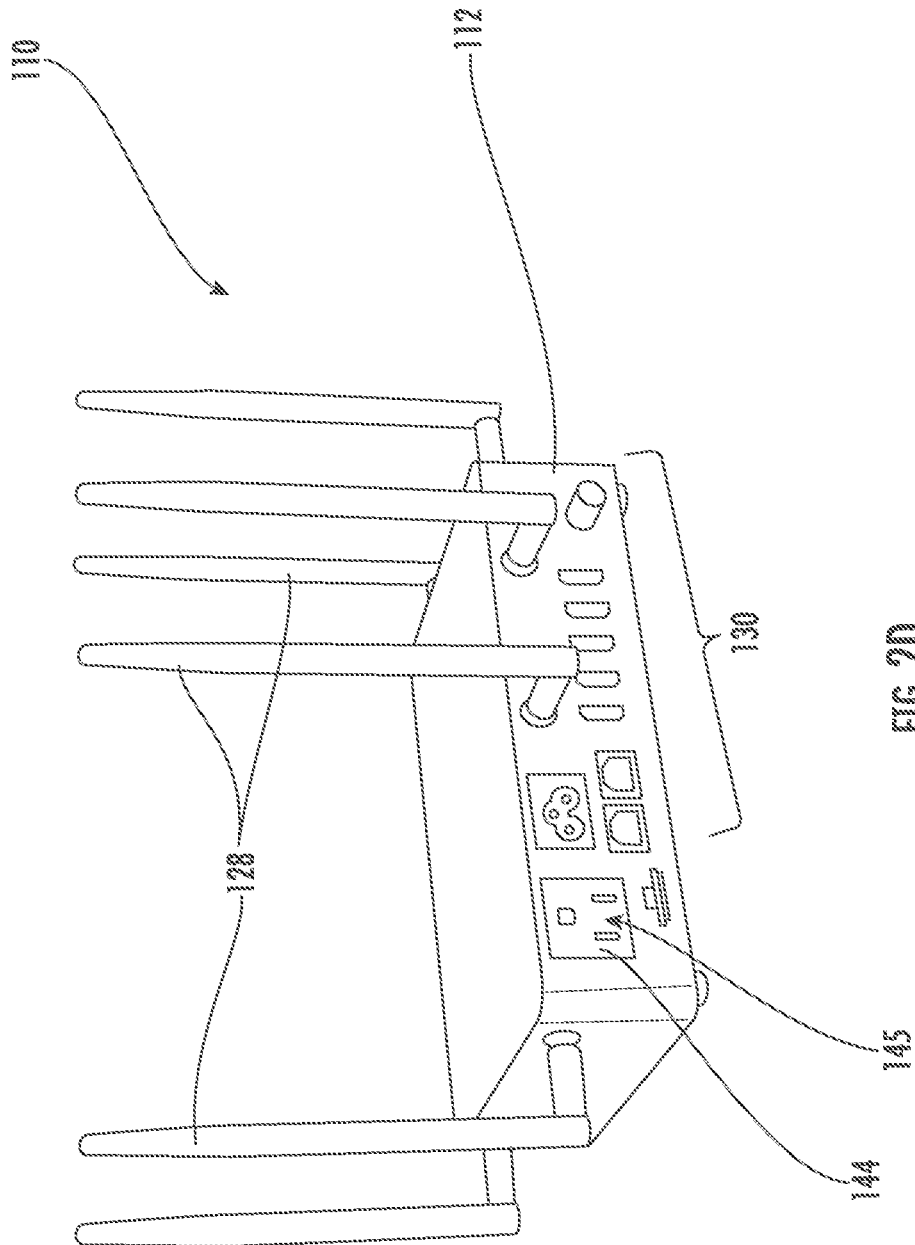
FIG. 2C

U.S. Patent

Feb. 23, 2021

Sheet 5 of 43

US 10,932,002 B2



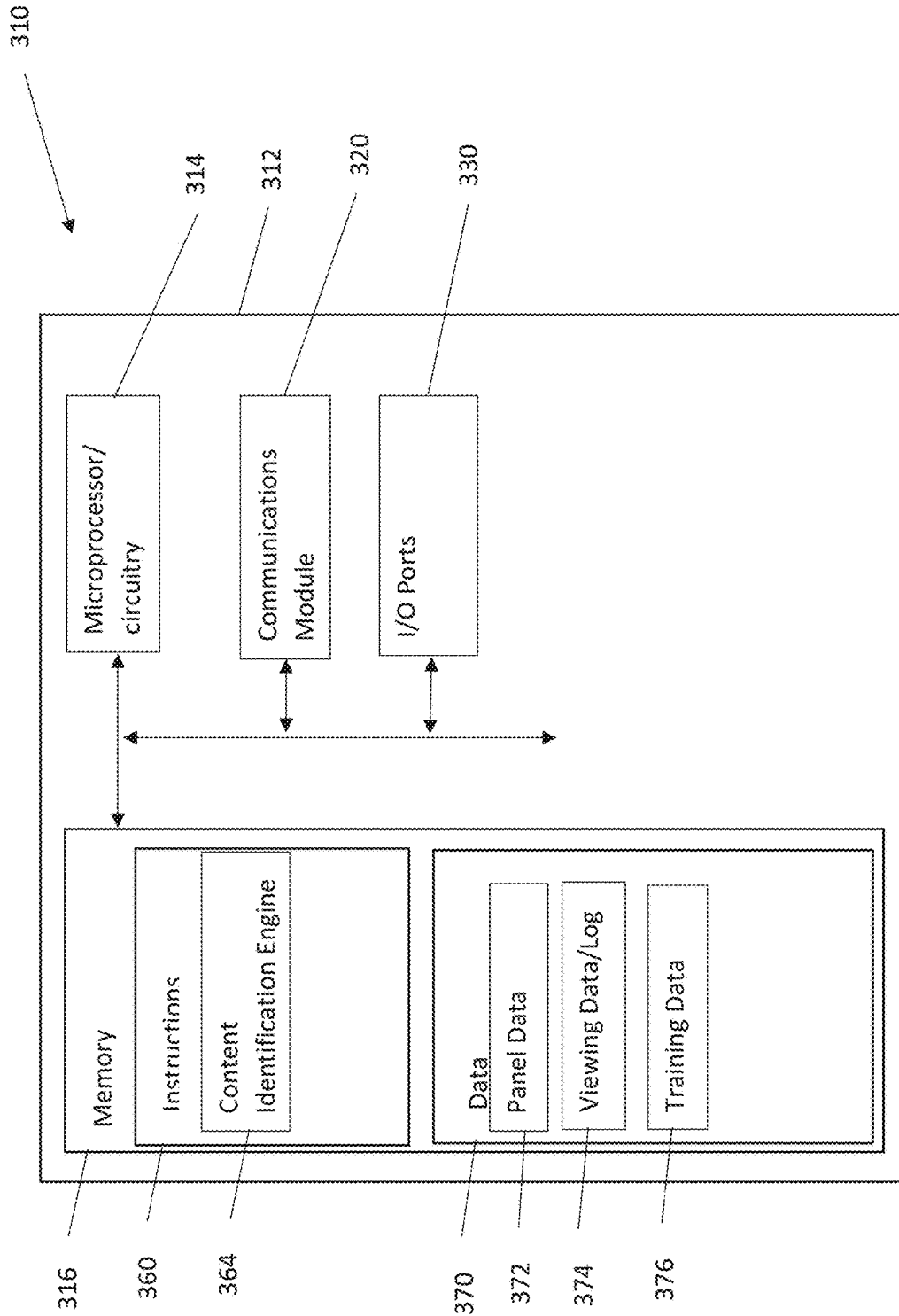


FIG. 3

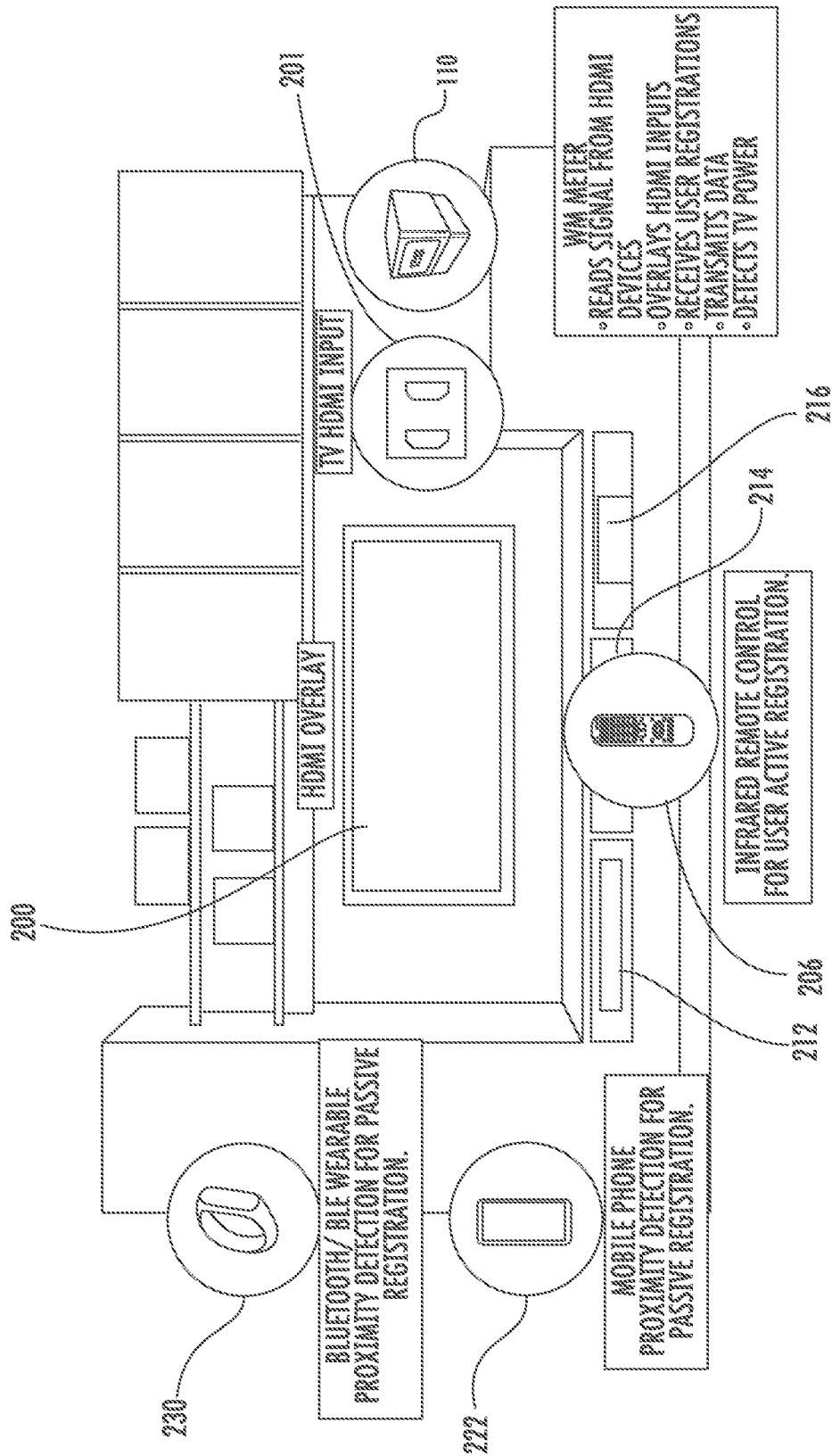


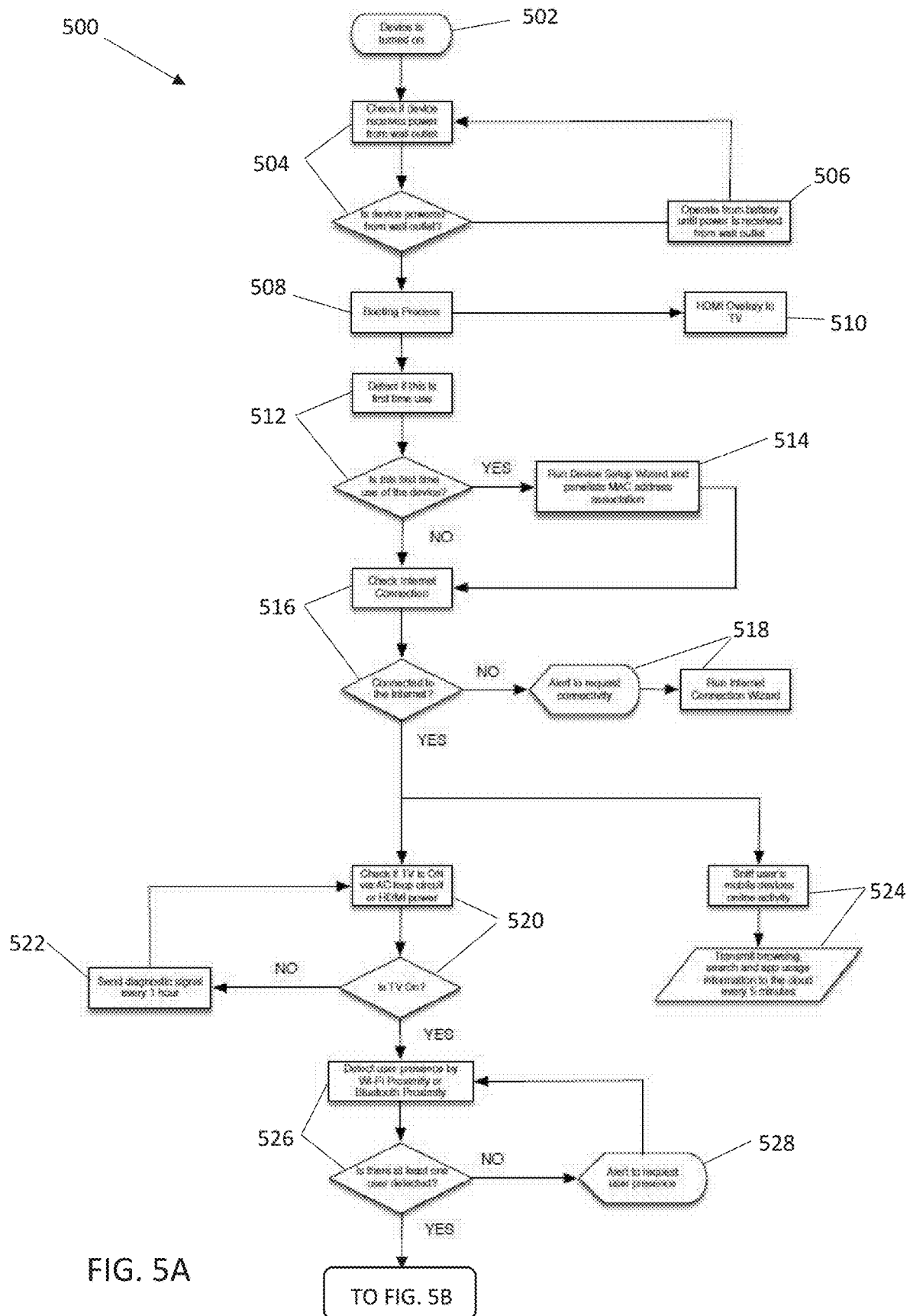
FIG. 4

U.S. Patent

Feb. 23, 2021

Sheet 8 of 43

US 10,932,002 B2



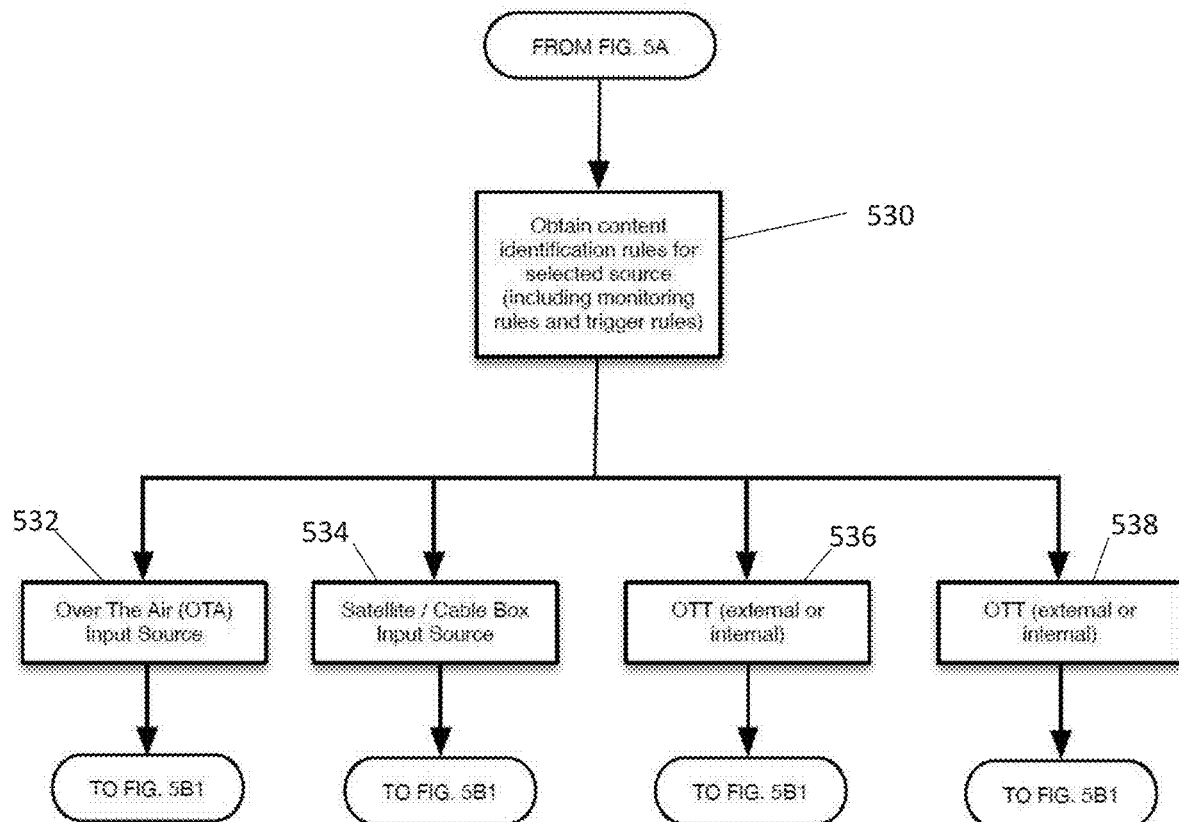


FIG. 5B

U.S. Patent

Feb. 23, 2021

Sheet 10 of 43

US 10,932,002 B2

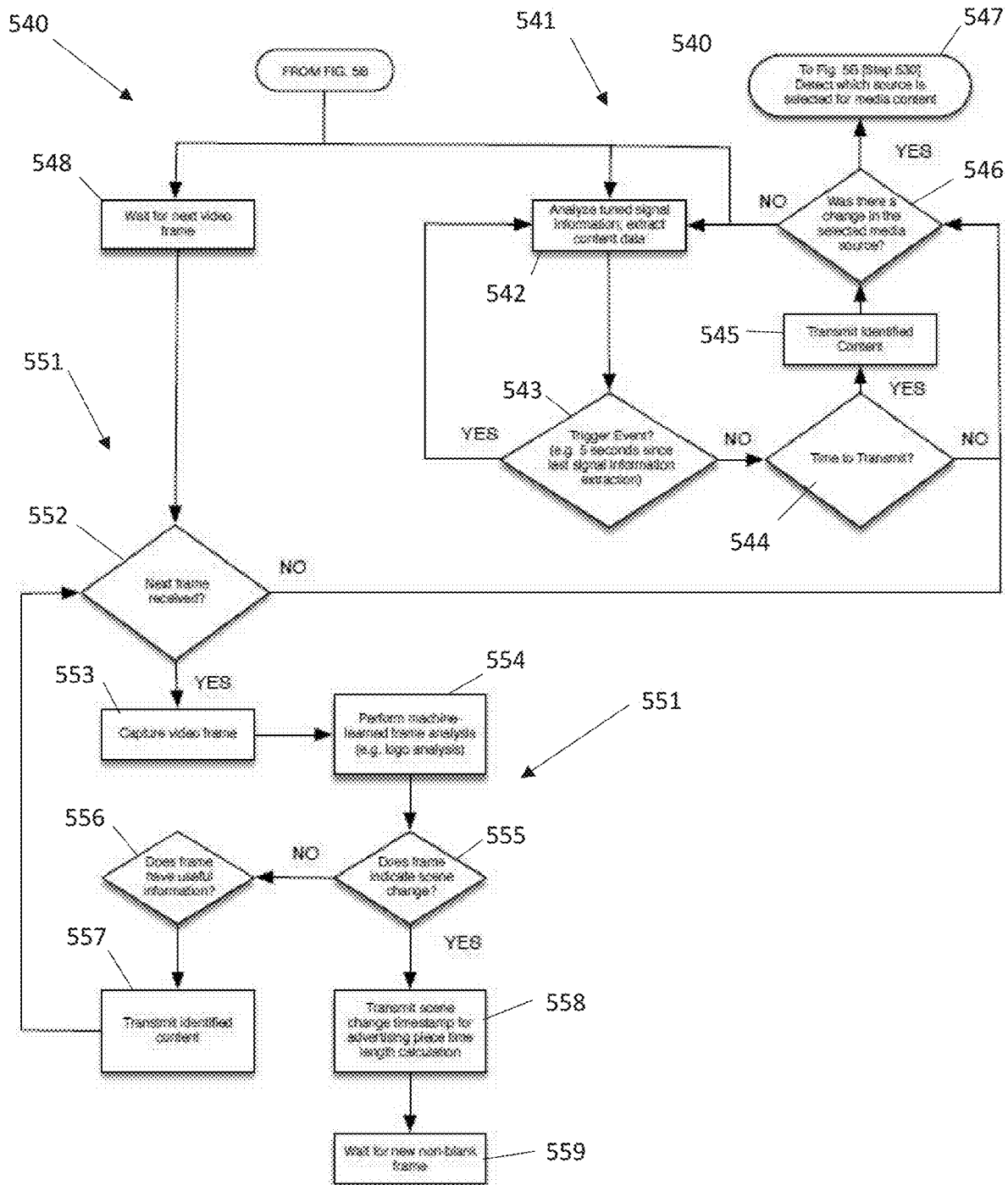
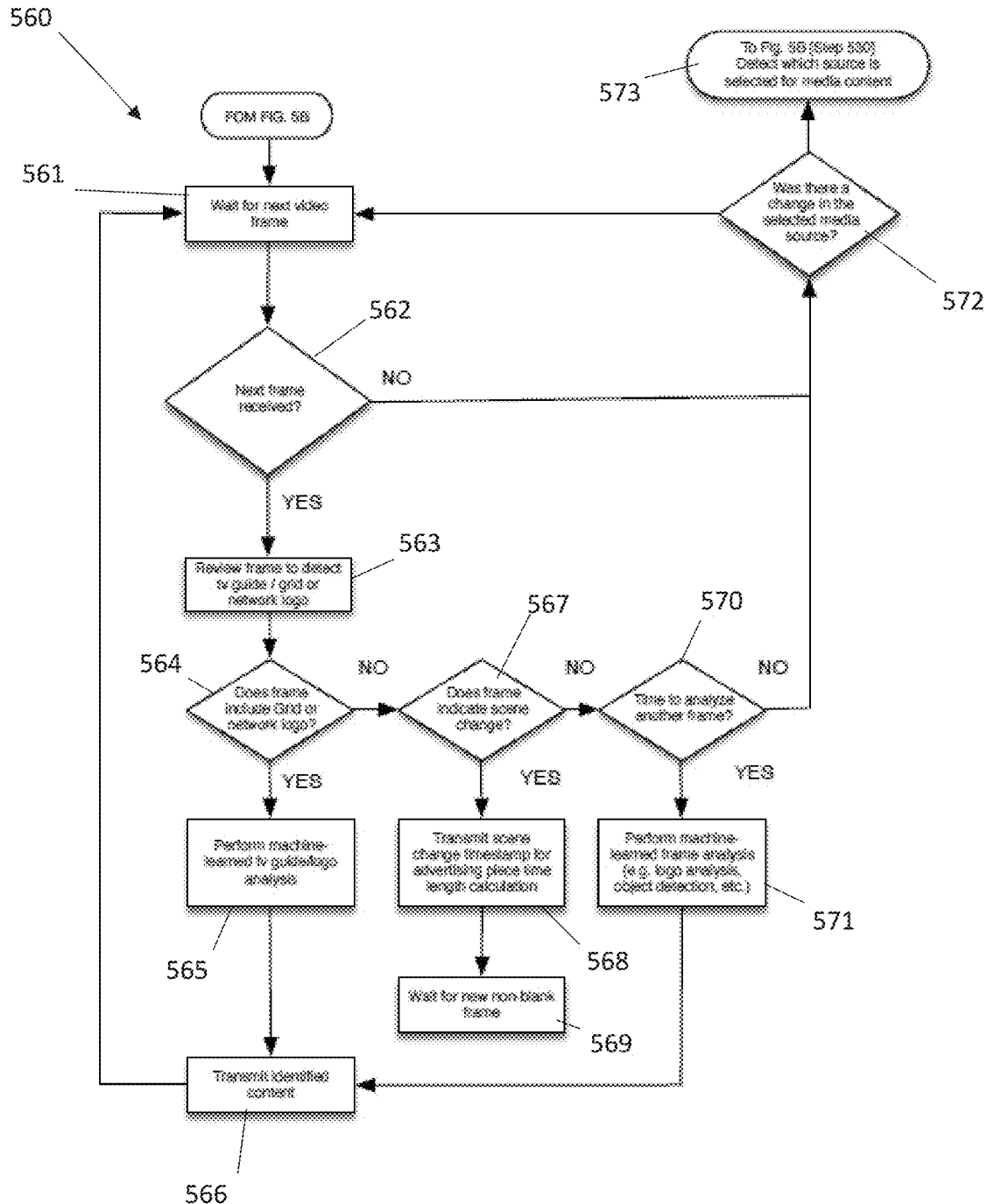


FIG. 5B1



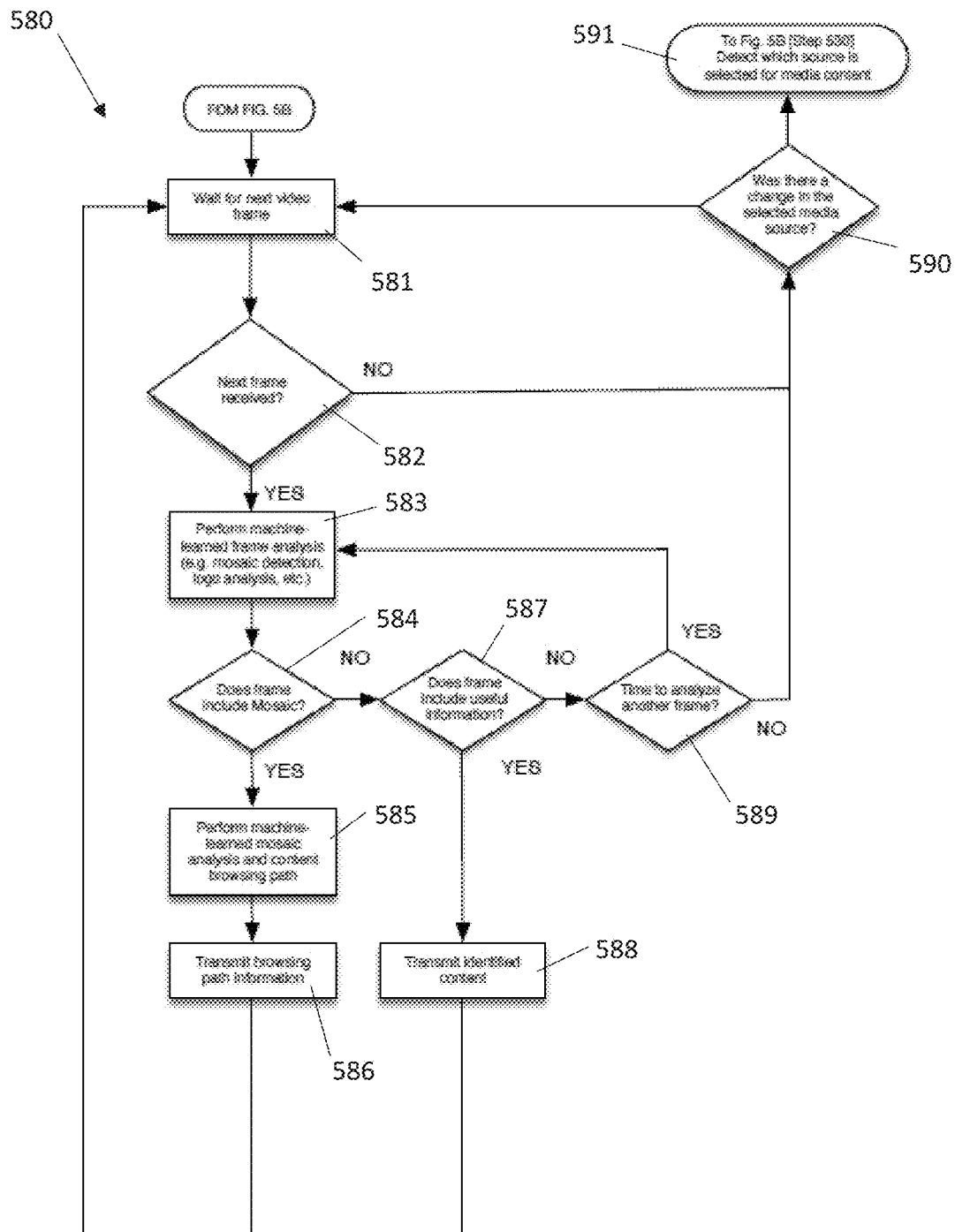


FIG. 5B3

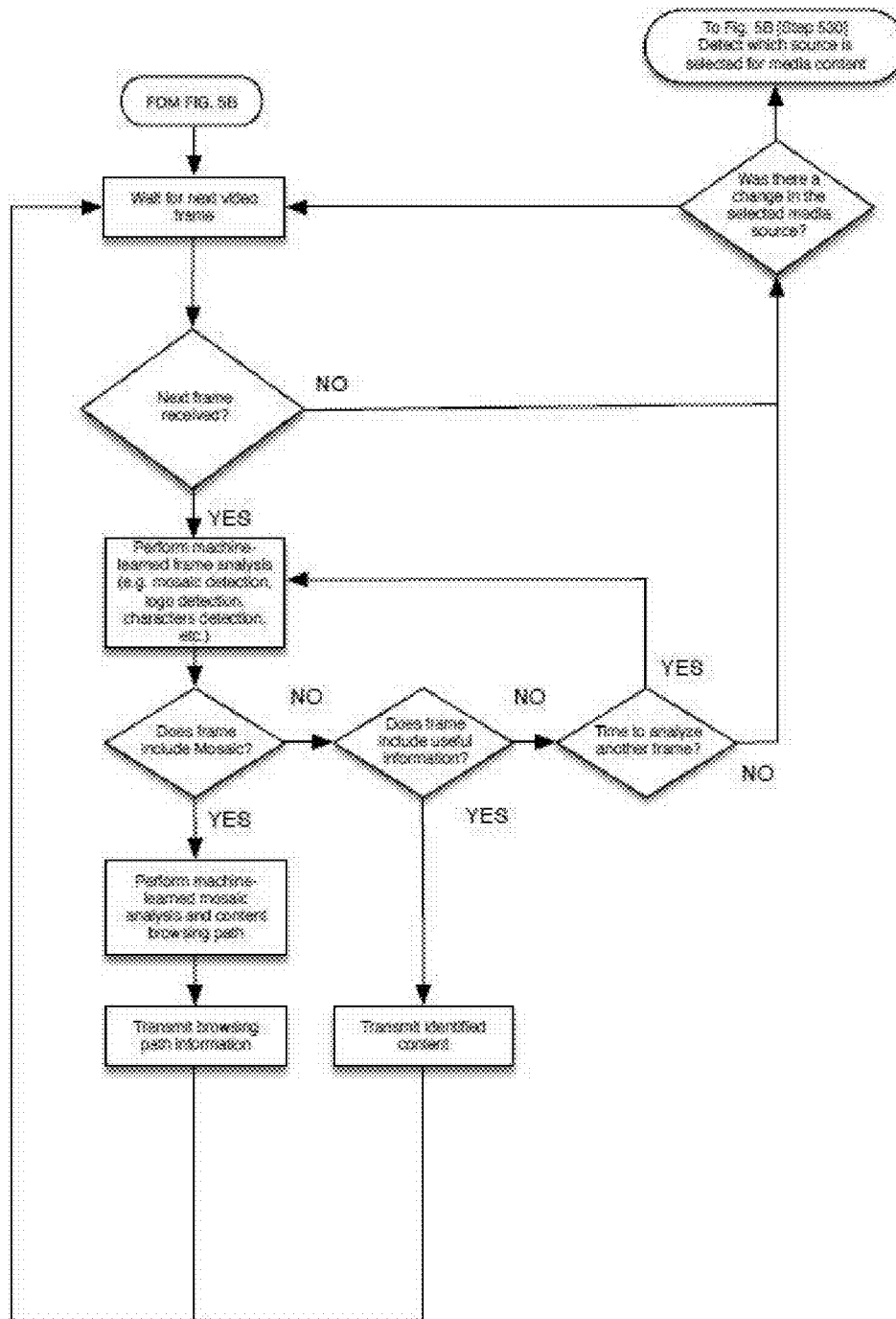


FIG. 5B4

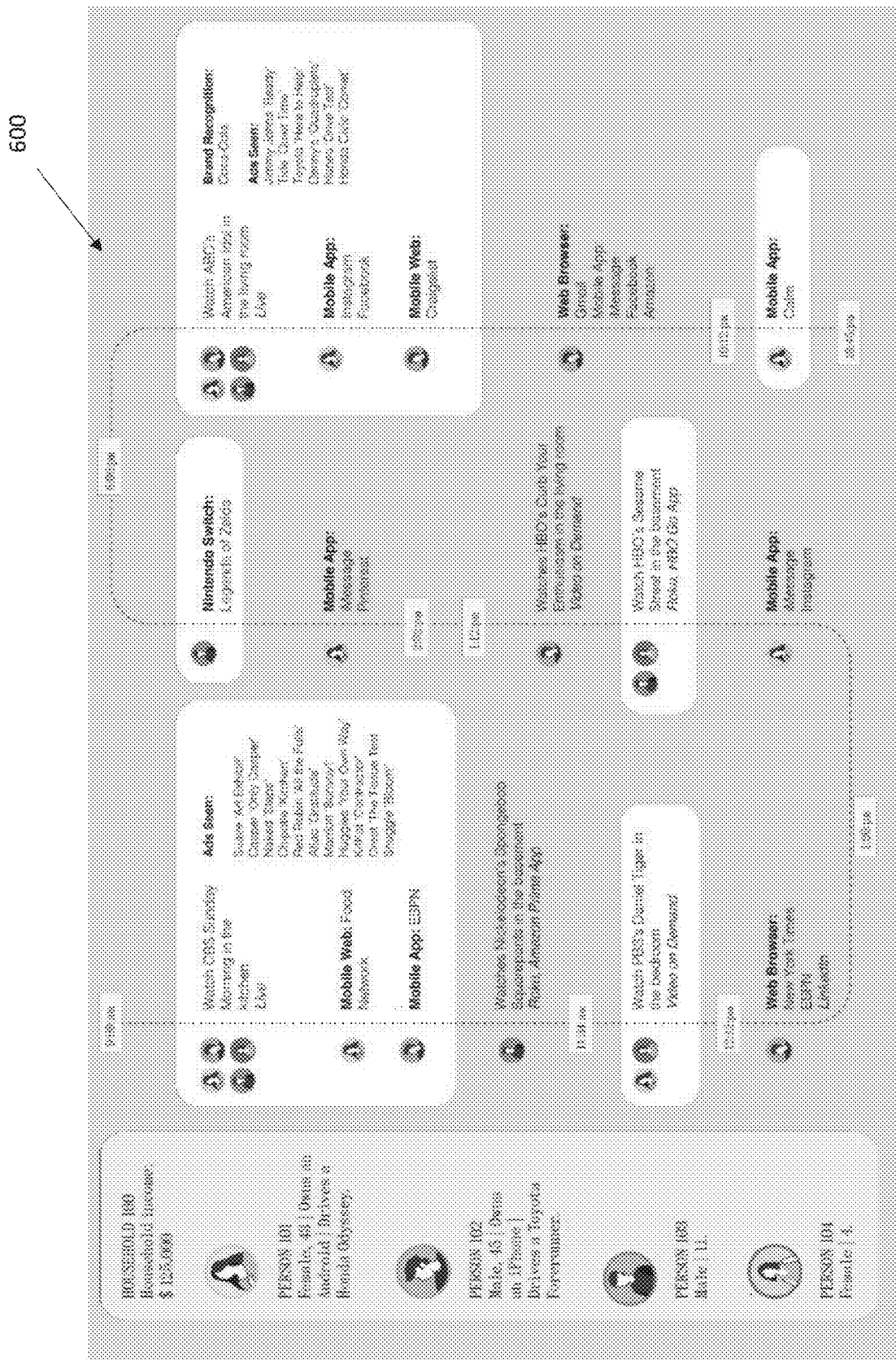


FIG. 6

700

152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------

751

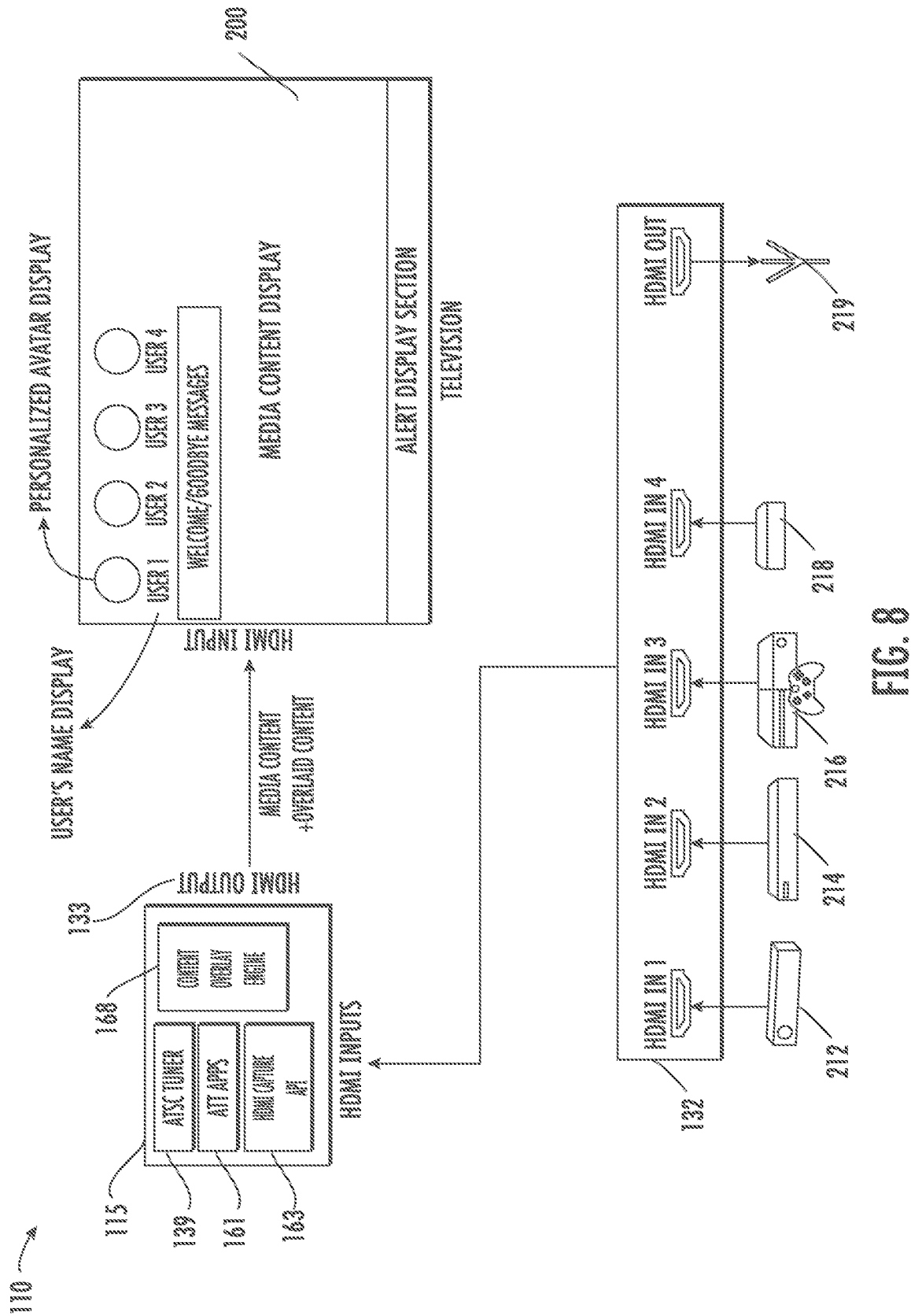
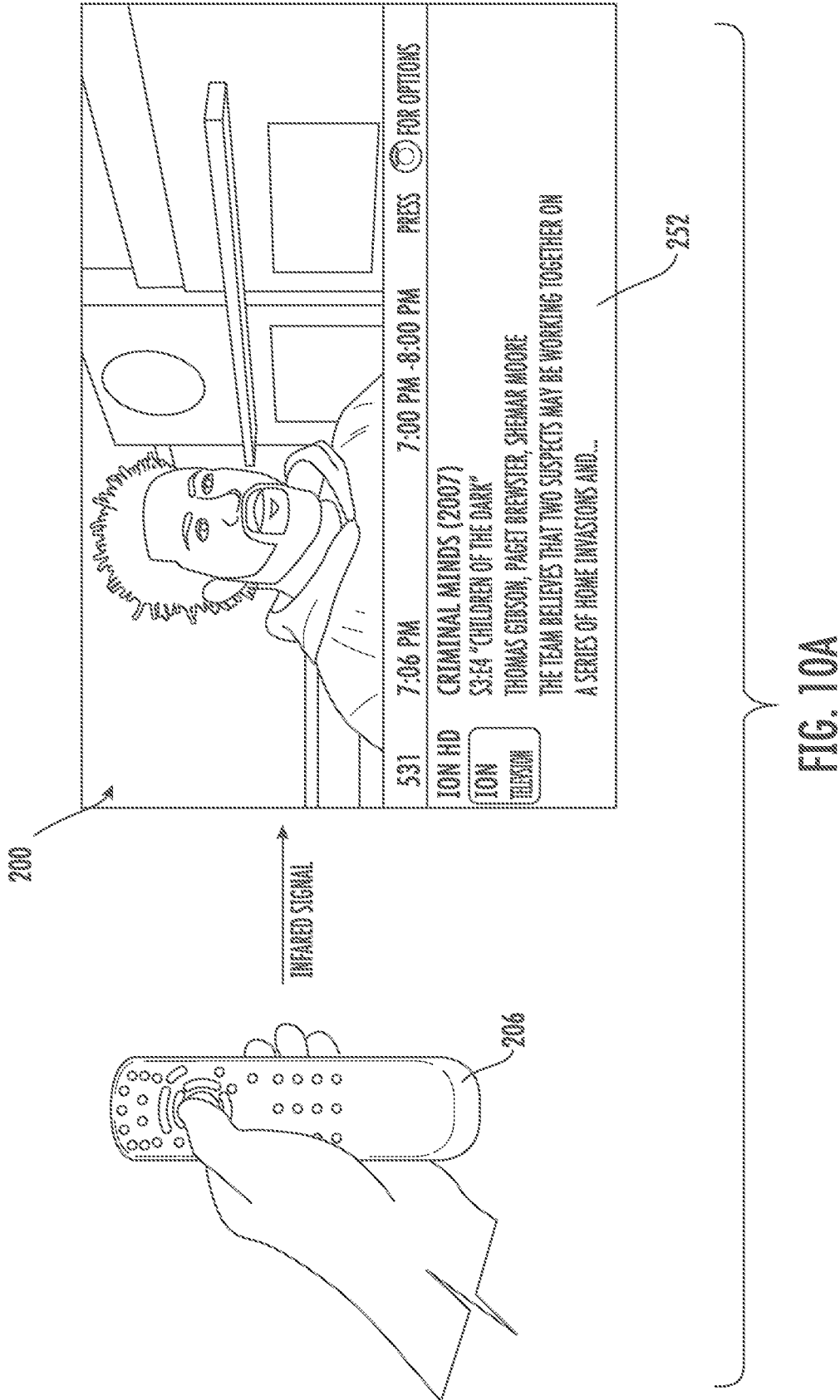


FIG. 8



FIG. 9



U.S. Patent

Feb. 23, 2021

Sheet 19 of 43

US 10,932,002 B2

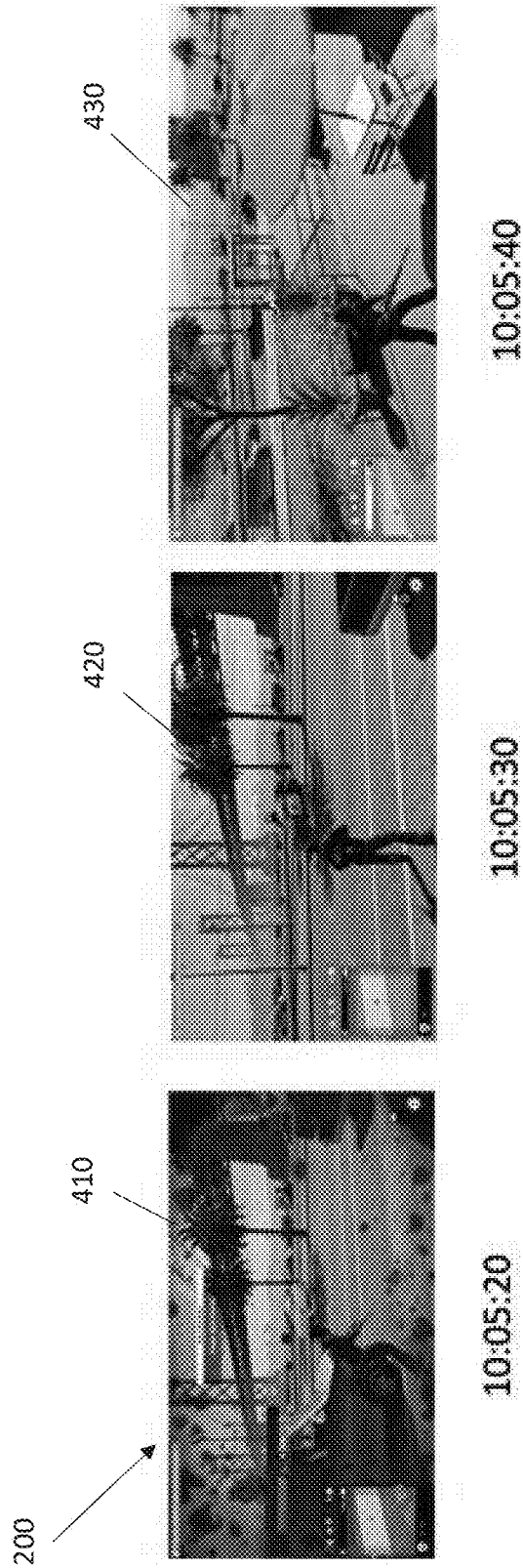
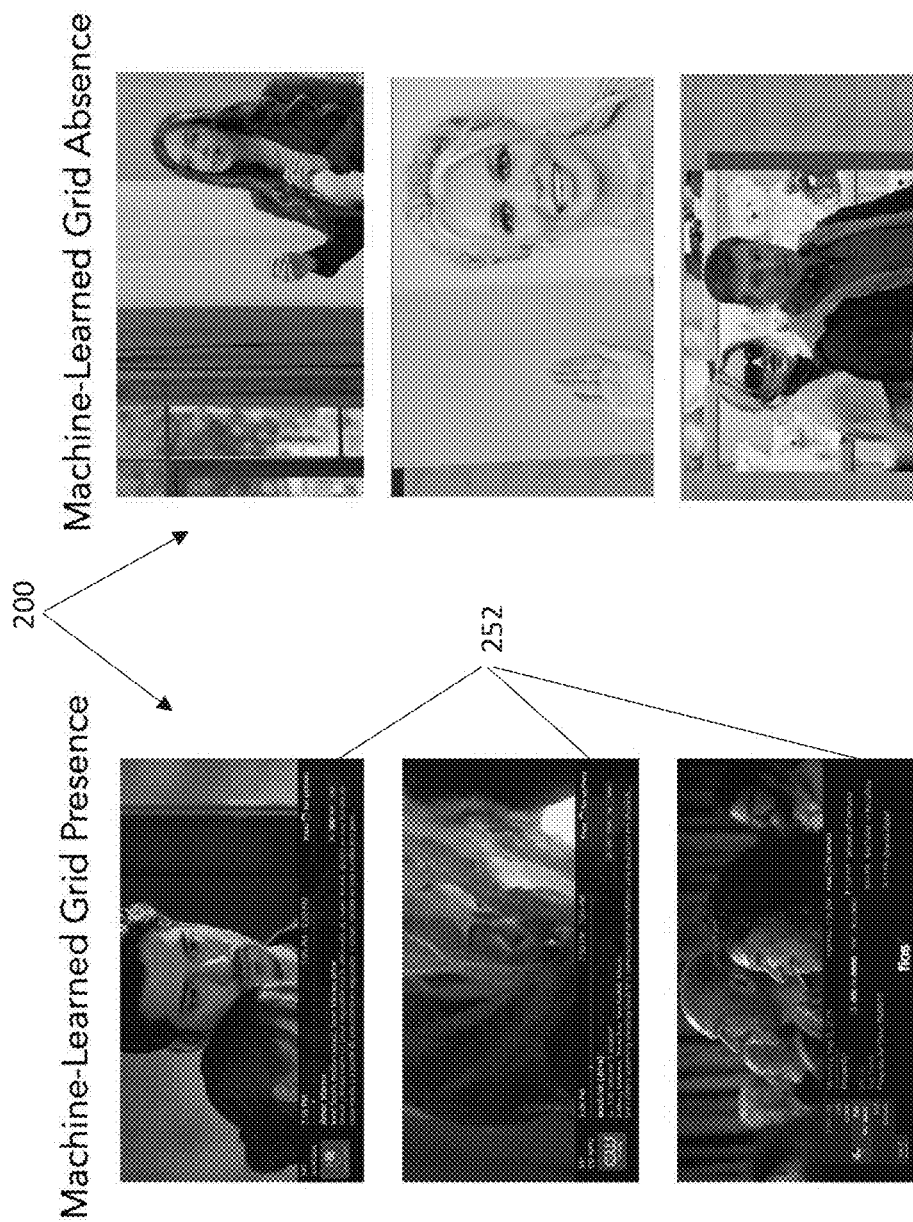


FIG. 10B



450

2020-06-12	19:06:13.196 5745-5745/com.personicore.hdmi1 I/MainActivity: ** Waiting for TV-guide/grid appearance **	451
2020-06-12	19:06:13.265 5745-5745/com.personicore.hdmi1 I/MainActivity: ** TV-guide/grid detected. Analyzing ...	452
2020-06-12	19:06:13.361 5745-5745/com.personicore.hdmi1 I/MainActivity: ** TV-guide/grid presence confirmed ...	453
2020-06-12	19:06:14.024 5745-5745/com.personicore.hdmi1 I/MainActivity: ** Extracting text from TV-guide/grid ...	454
2020-06-12	19:06:14.120 5745-5745/com.personicore.hdmi1 I/MainActivity: ** Extracting program: Criminal Minds (2007) **	455
2020-06-12	19:06:14.191 5745-5745/com.personicore.hdmi1 I/MainActivity: ** Extracting network: ion HD **	456
2020-06-12	19:06:14.263 5745-5745/com.personicore.hdmi1 I/MainActivity: ** Extracting channel number: 531 ...	457
2020-06-12	19:06:14.196 5745-5745/com.personicore.hdmi1 I/MainActivity: ** Transmitting data to server ...	458
2020-06-12	19:06:14.328 5745-5745/com.personicore.hdmi1 I/MainActivity: ** Waiting for TV-guide/grid appearance **	459

FIG. 10D

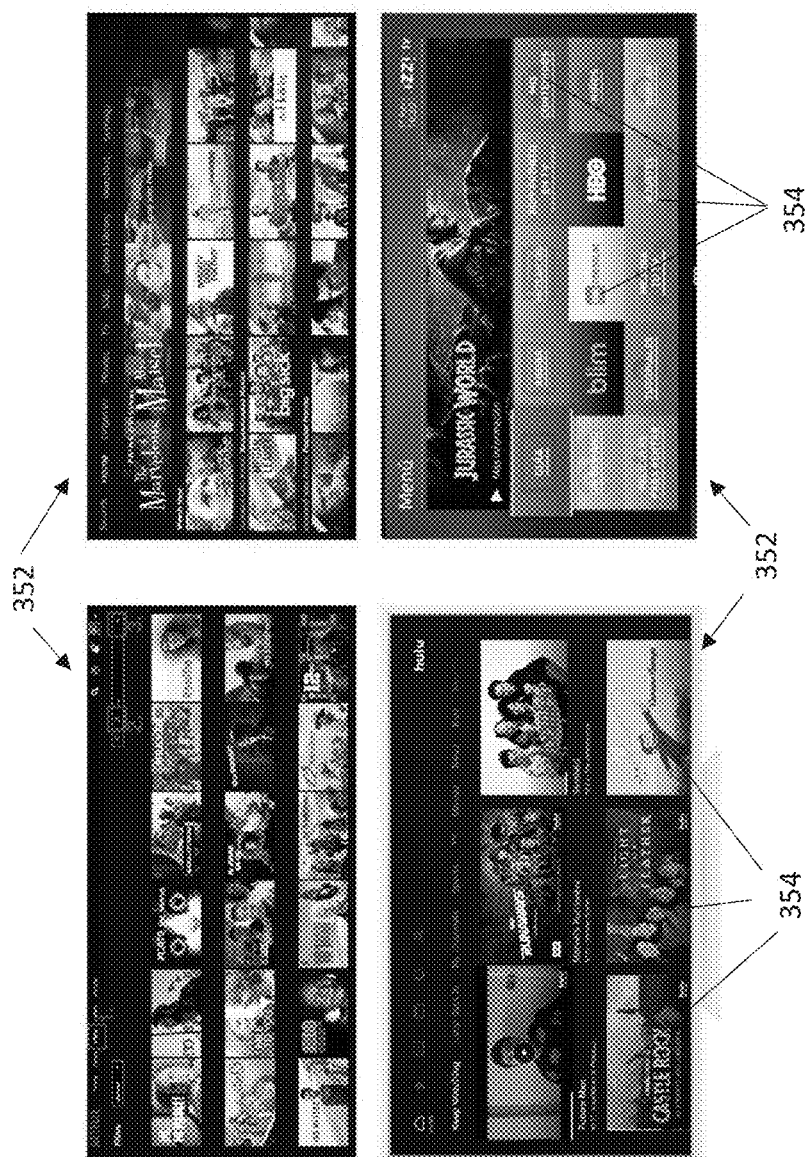


FIG. 10E

460

2020-06-12	20:05:18.014	5745-5745/com.personicore.hdmi3 I/MainActivity: ** Waiting for content grid appearance **	461
2020-06-12	20:05:18.111	5745-5745/com.personicore.hdmi3 I/MainActivity: ** Content grid detected. Analyzing ...	462
2020-06-12	20:05:18.435	5745-5745/com.personicore.hdmi3 I/MainActivity: ** Content grid presence confirmed ...	463
2020-06-12	20:06:19.216	5745-5745/com.personicore.hdmi3 I/MainActivity: ** Extracting text from the content grid ...	464
2020-06-12	20:06:19.308	5745-5745/com.personicore.hdmi3 I/MainActivity: ** Analyzing the selected option ...	465
2020-06-12	20:06:19.308	5745-5745/com.personicore.hdmi3 I/MainActivity: ** Extracting text from selected option....	466
2020-06-12	20:06:19.354	5745-5745/com.personicore.hdmi3 I/MainActivity: ** Selected option: TV **	467
2020-06-12	20:06:19.481	5745-5745/com.personicore.hdmi3 I/MainActivity: ** Transmitting data to server ...	468
2020-06-12	20:06:19.495	5745-5745/com.personicore.hdmi3 I/MainActivity: ** Waiting for content grid appearance **	469

FIG. 10F

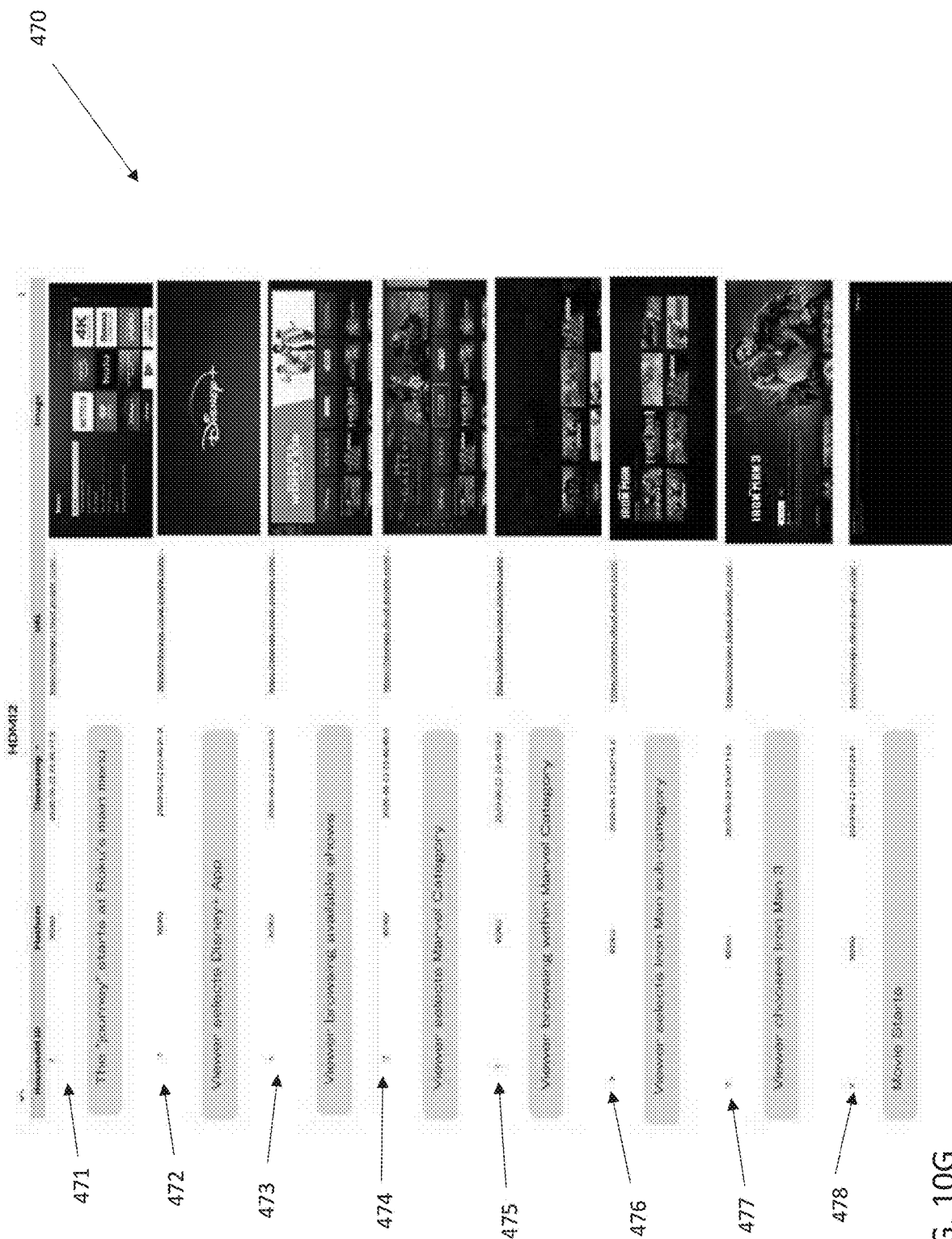


FIG. 10G

U.S. Patent

Feb. 23, 2021

Sheet 25 of 43

US 10,932,002 B2

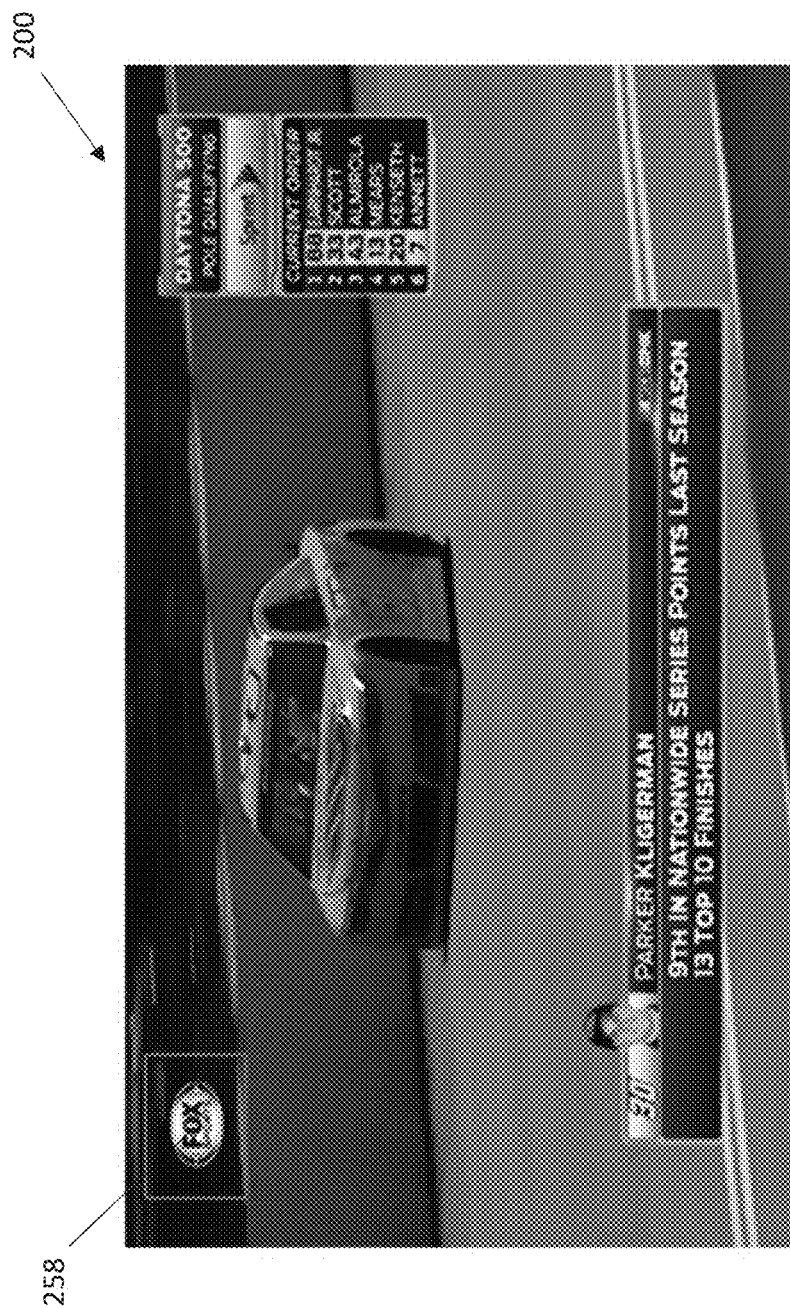


FIG. 10H

480

2020-06-16	07:35:13.046 5745-5745/com.personicore.hdmi1 I/MainActivity: ** Detecting logo presence **	481
2020-06-16	07:35:13.189 5745-5745/com.personicore.hdmi1 I/MainActivity: ** Network logo detected. Analyzing ... **	482
2020-06-16	07:35:14.180 5745-5745/com.personicore.hdmi1 I/MainActivity: ** Logo recognized: FOX SPORTS **	483
2020-06-16	07:36:14.394 5745-5745/com.personicore.hdmi1 I/MainActivity: ** Transmitting data to server ... **	484
2020-06-16	07:36:14.401 5745-5745/com.personicore.hdmi1 I/MainActivity: ** Detecting logo presence **	485
2020-06-16	07:48:54.238 5745-5745/com.personicore.hdmi1 I/MainActivity: ** Logo no longer present **	486
2020-06-16	07:48:54.394 5745-5745/com.personicore.hdmi1 I/MainActivity: ** Transmitting data to server ... **	487

Fig. 10I

U.S. Patent

Feb. 23, 2021

Sheet 27 of 43

US 10,932,002 B2

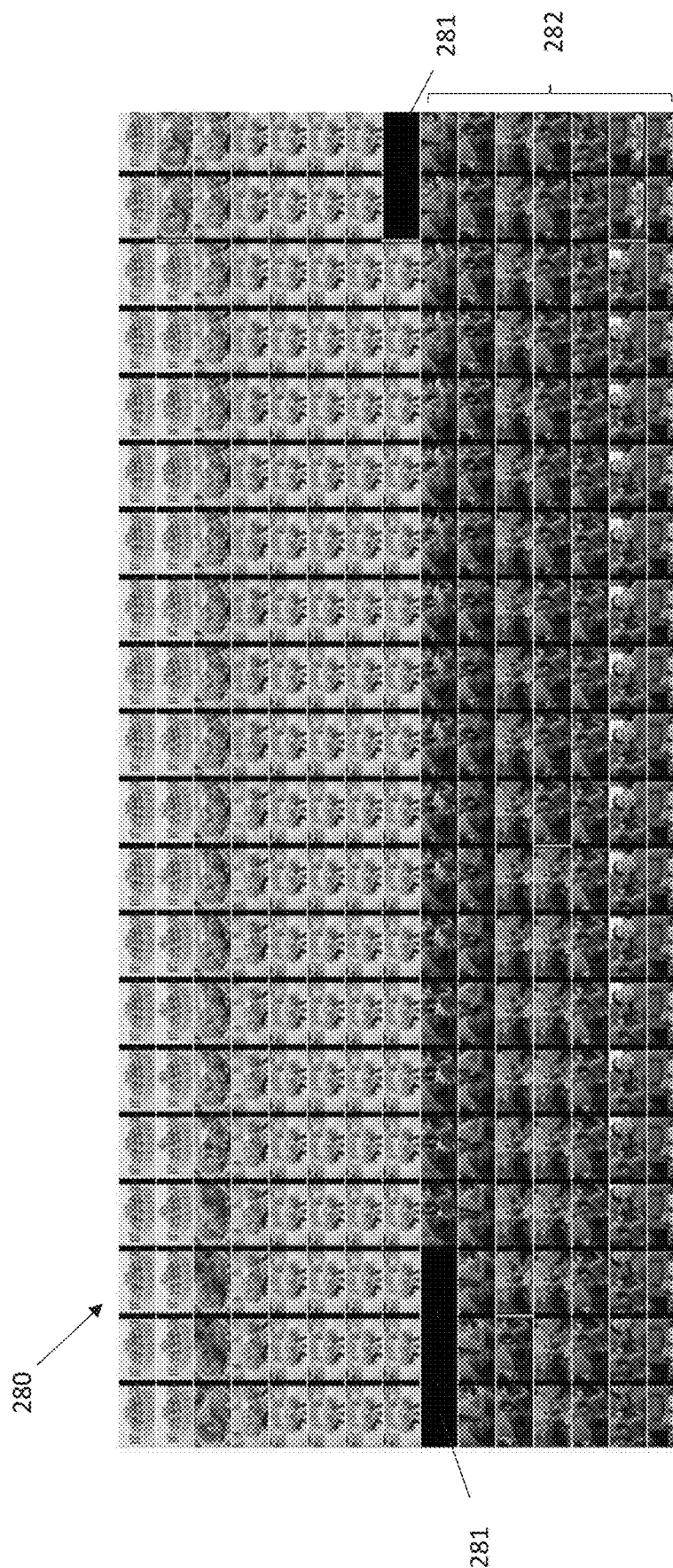
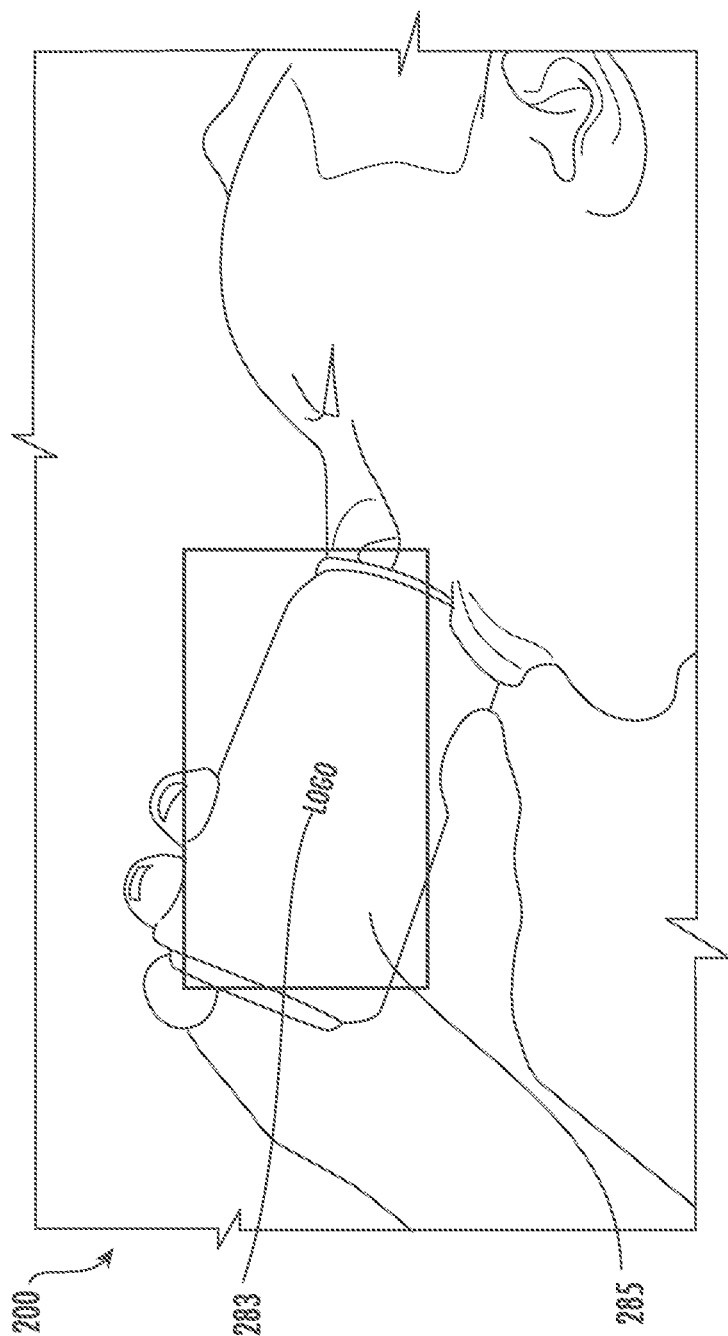
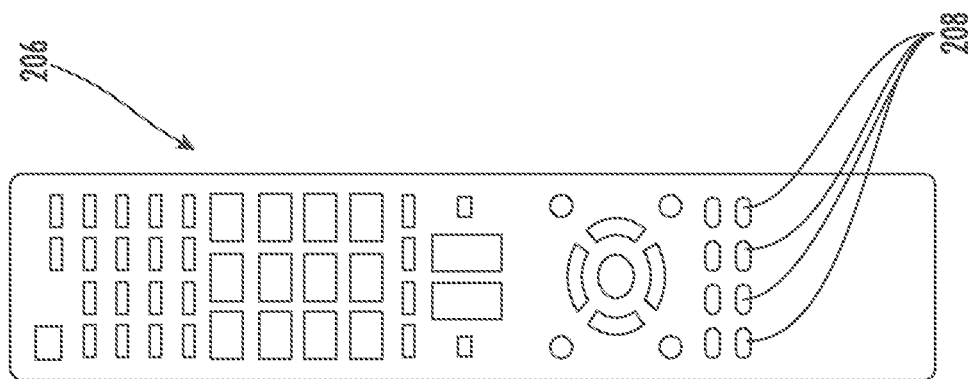


FIG. 10J



1710

CH 9 || Elapsed: 1 min || 2007-04-26 17:41 || WPA handshake: 00:14:6C:7E:40:80

1720

BSSID	PWR	RXQ	Beacons	#Data	#/s	CH	MB	ENC	CIPHER	AUTH	ESSID
00:09:5B:1C:AA:1D	11	16	10	0	0	11	54	OPN			NETGEAR
00:14:6C:7A:41:81	34	100	57	14	1	9	11e	WEP	WEP		bigbear
00:14:6C:7E:40:80	32	100	752	73	2	9	54	WPA	TKIP	P5K	teddy

1700

BSSID	STATION	PWR	Rate	Lost	Packets	Probes
00:14:6C:7A:41:81	00:0F:85:32:31:31	51	36-24	2	14	
(not associated)	00:14:A4:3F:8D:13	19	0-0	0	4	mossy
00:14:6C:7A:41:81	00:0C:41:52:D1:D1	-1	36-36	0	5	
00:14:6C:7E:40:80	00:0F:85:FD:FB:C2	35	54-54	0	99	teddy

FIG. 12

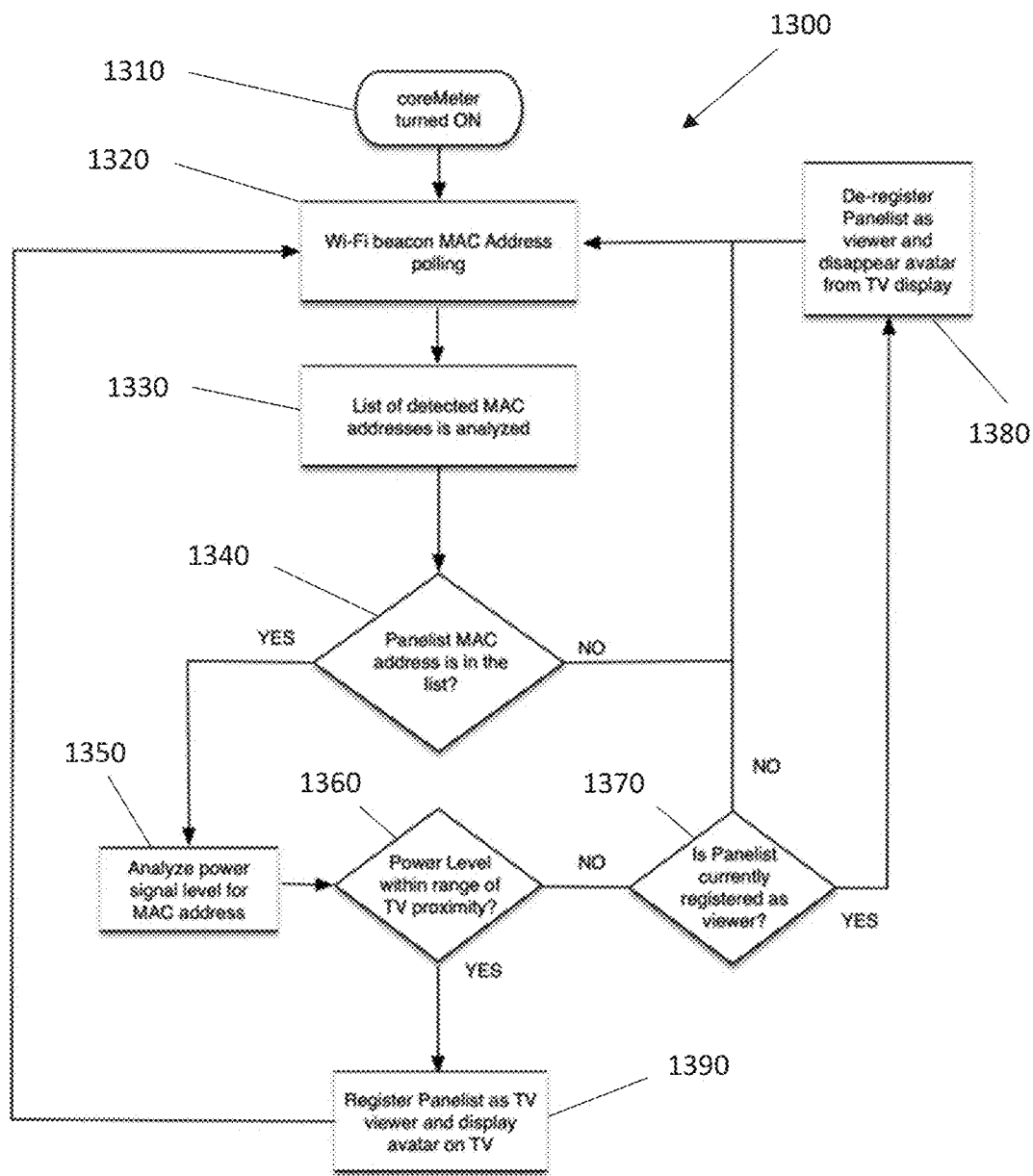


FIG. 13

U.S. Patent

Feb. 23, 2021

Sheet 31 of 43

US 10,932,002 B2

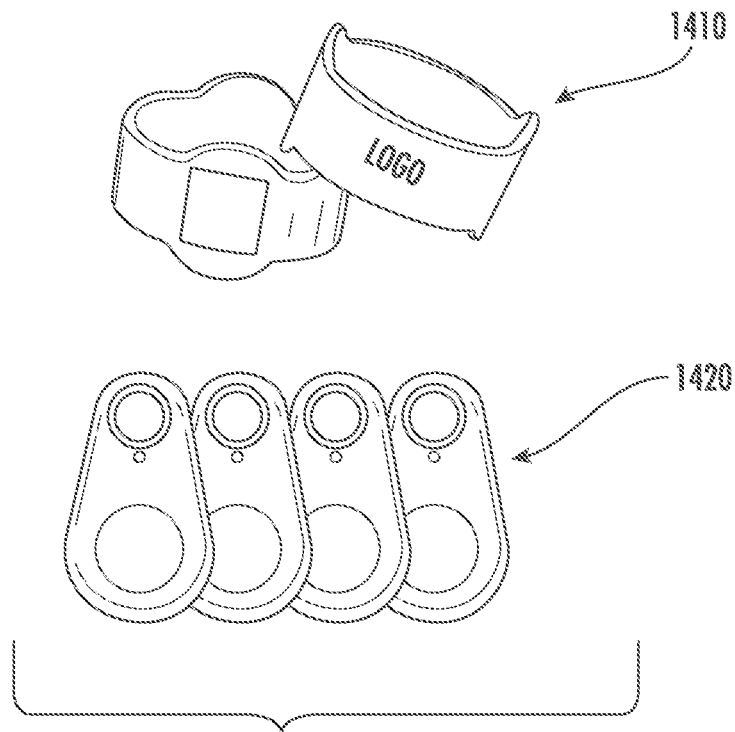


FIG. 14

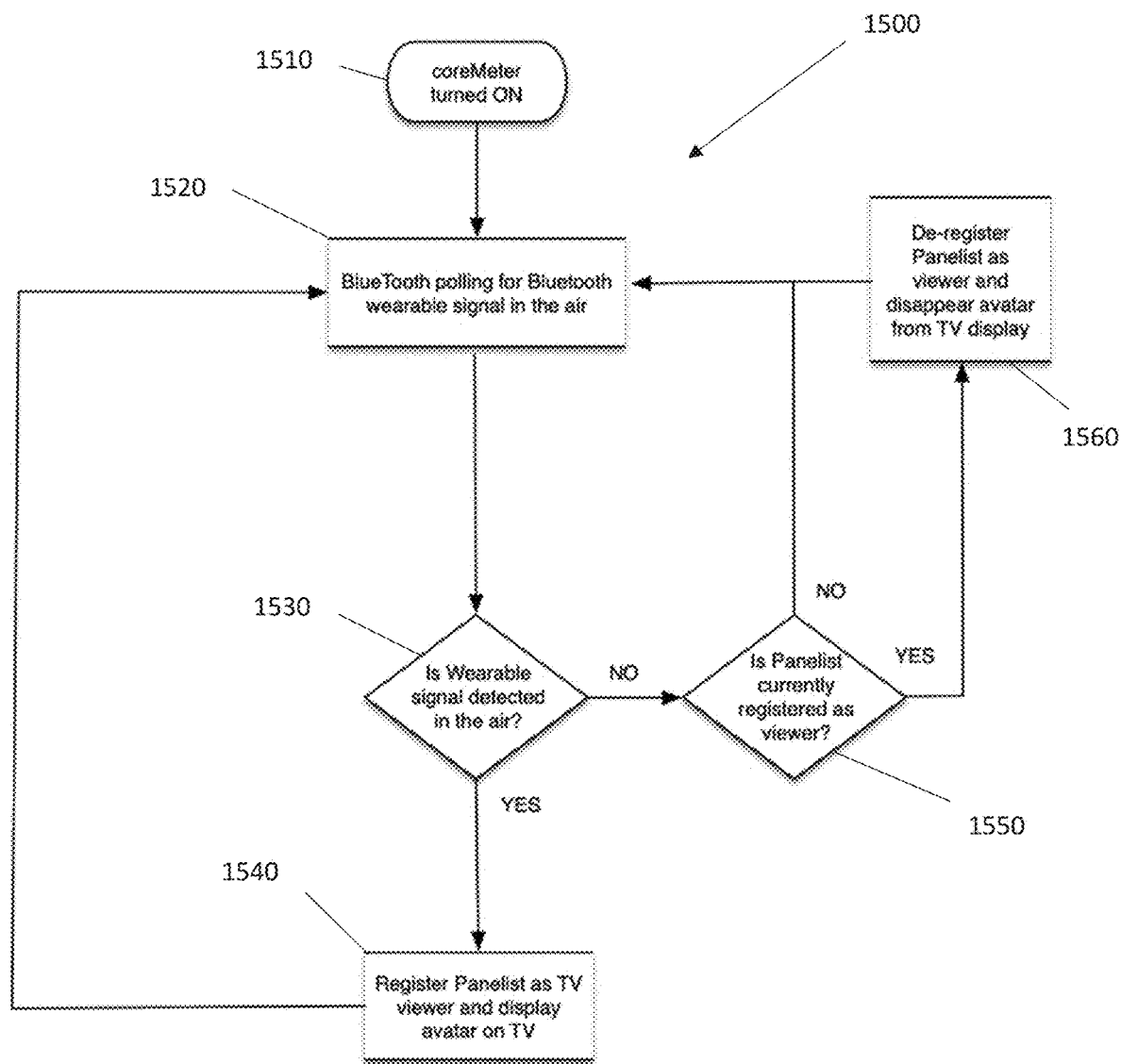


FIG. 15

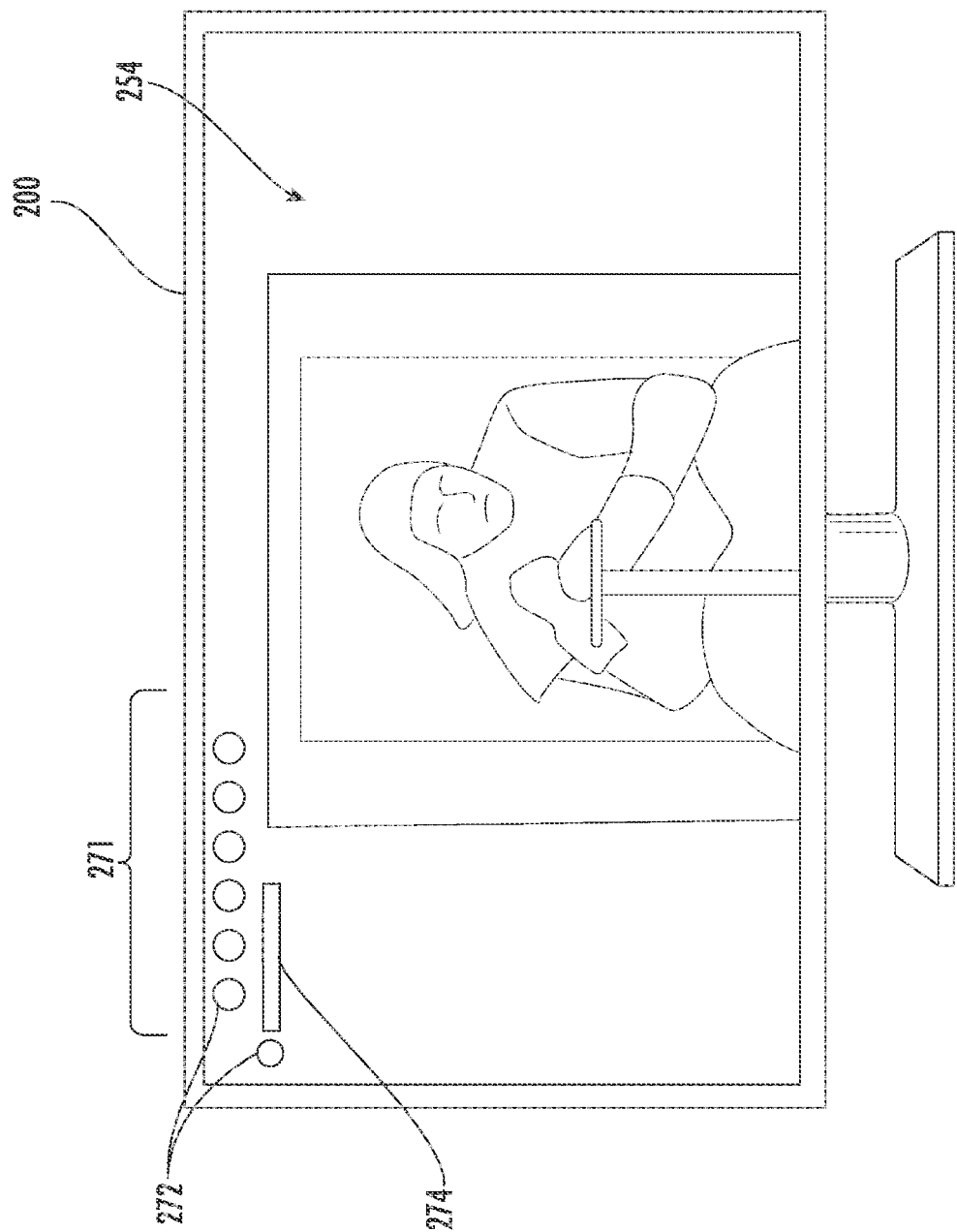
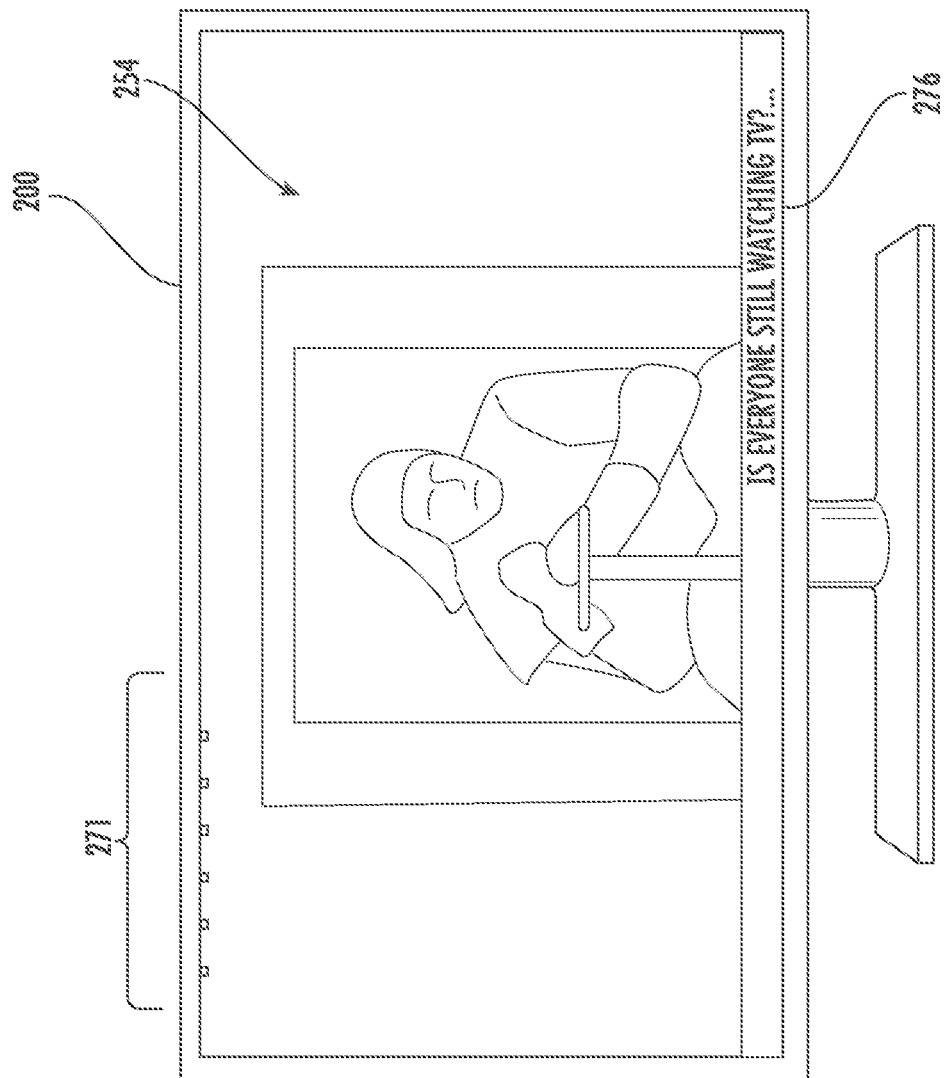


FIG. 16



1800



```
GET https://github.com/
~ 200 text/html 5.52kB
GET https://a248.e.akamai.net/assets.github.com/stylesheets/bundles/github2-24f559e3ded11f2a
1c7ef9ee730882bd8d550c7b8.css
~ 200 text/css 28.27kB
GET https://a248.e.akamai.net/assets.github.com/images/modules/header/loginv2@4x-hover.png?1
324325424
~ 200 image/png 6.01kB
GET https://a248.e.akamai.net/assets.github.com/javascripts/bundles/jquery-b2ca87cb3c906ce2
cfd58811b430b8bc25245926.js
~ 200 application/x-javascript 32.59kB
GET https://a248.e.akamai.net/assets.github.com/stylesheets/bundles/github-cb564c47c51a14
af1ae265d7ebab59c4e78b92cb.css
~ 200 text/css 37.09kB
GET https://a248.e.akamai.net/assets.github.com/images/modules/home/logos/facebook.png?1324
526958
~ 200 image/png 5.55kB
>> GET https://github.com/twitter
```

1810

FIG. 18

U.S. Patent

Feb. 23, 2021

Sheet 36 of 43

US 10,932,002 B2

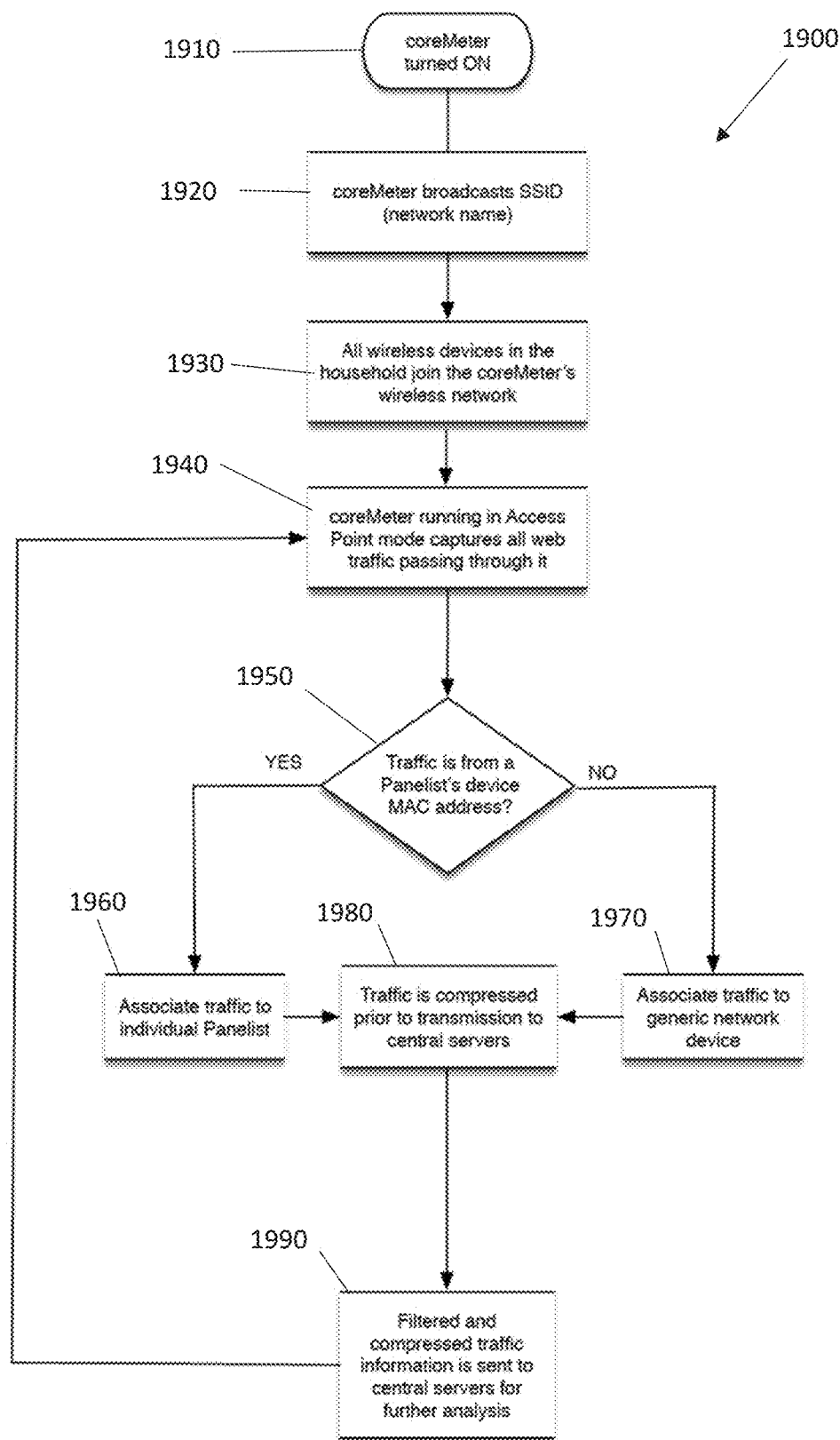


FIG. 19

U.S. Patent

Feb. 23, 2021

Sheet 37 of 43

US 10,932,002 B2

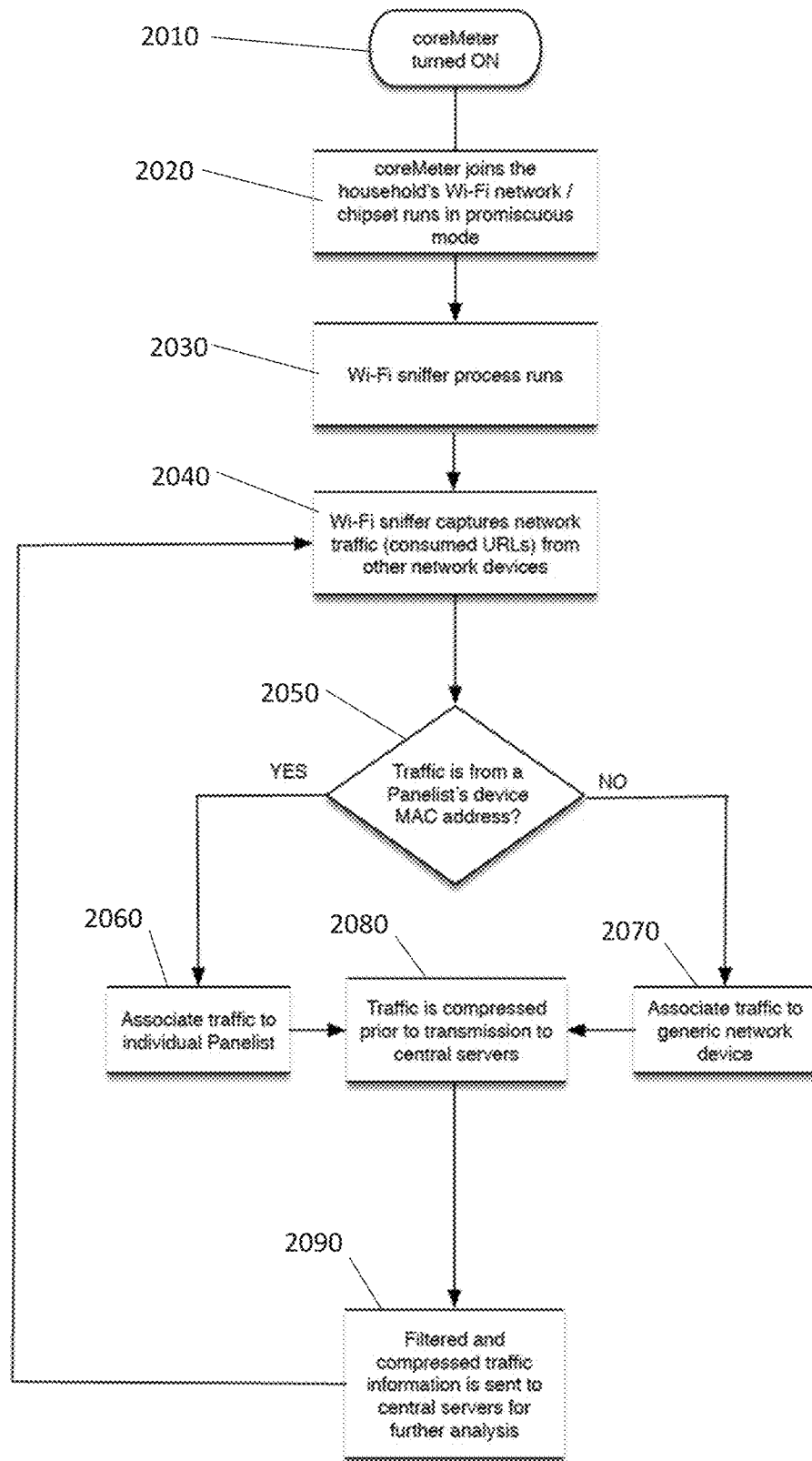


FIG. 20

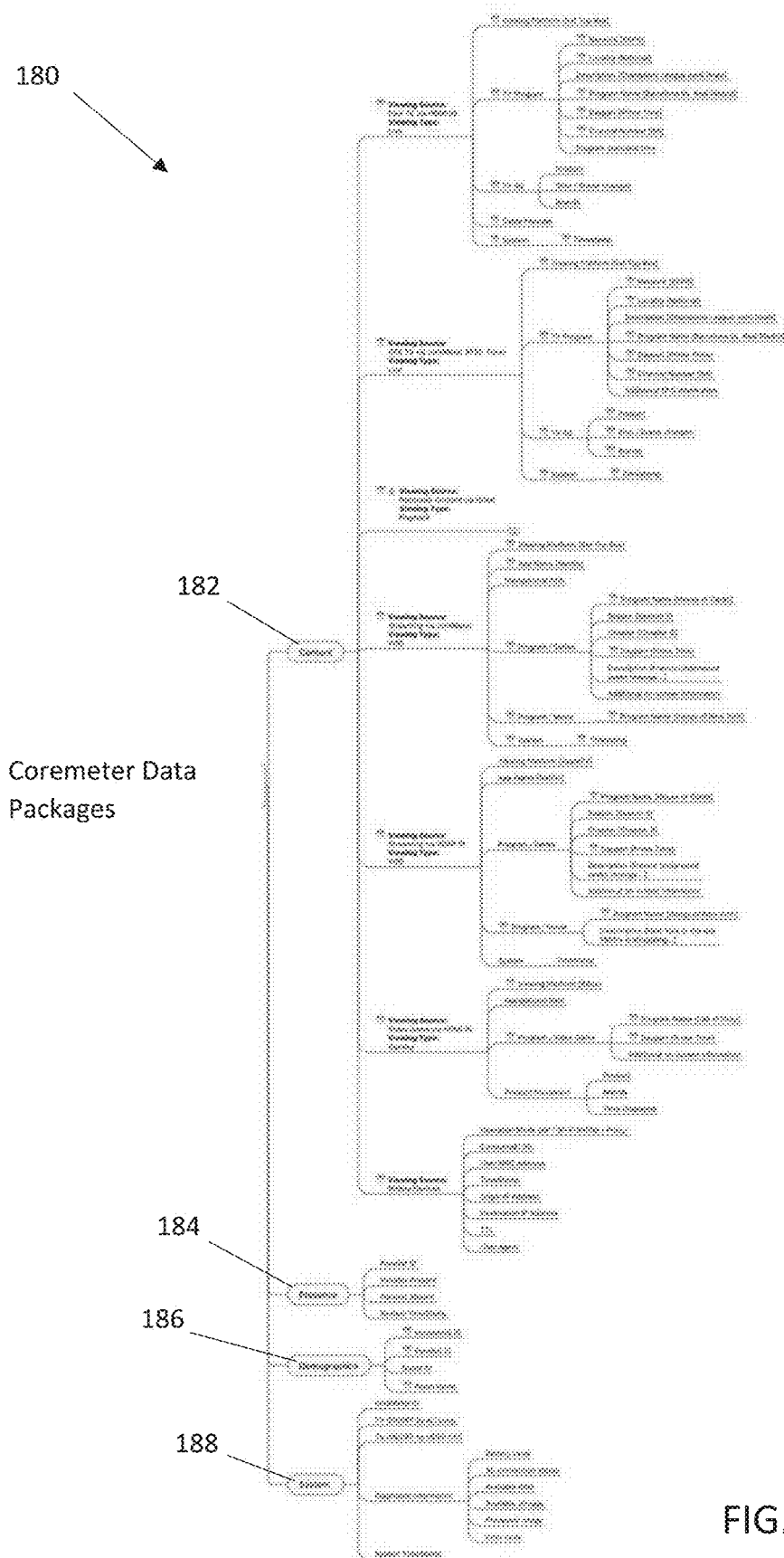


FIG. 21BFIG. 21C

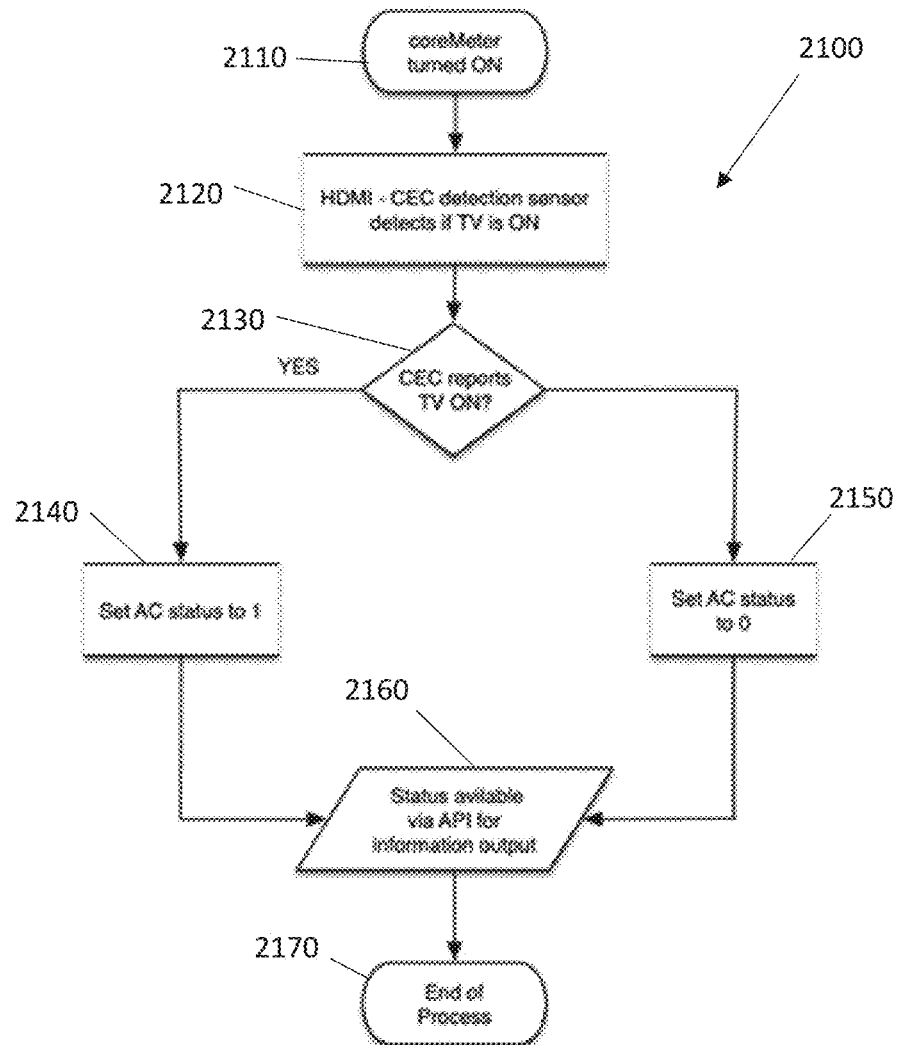


FIG. 22

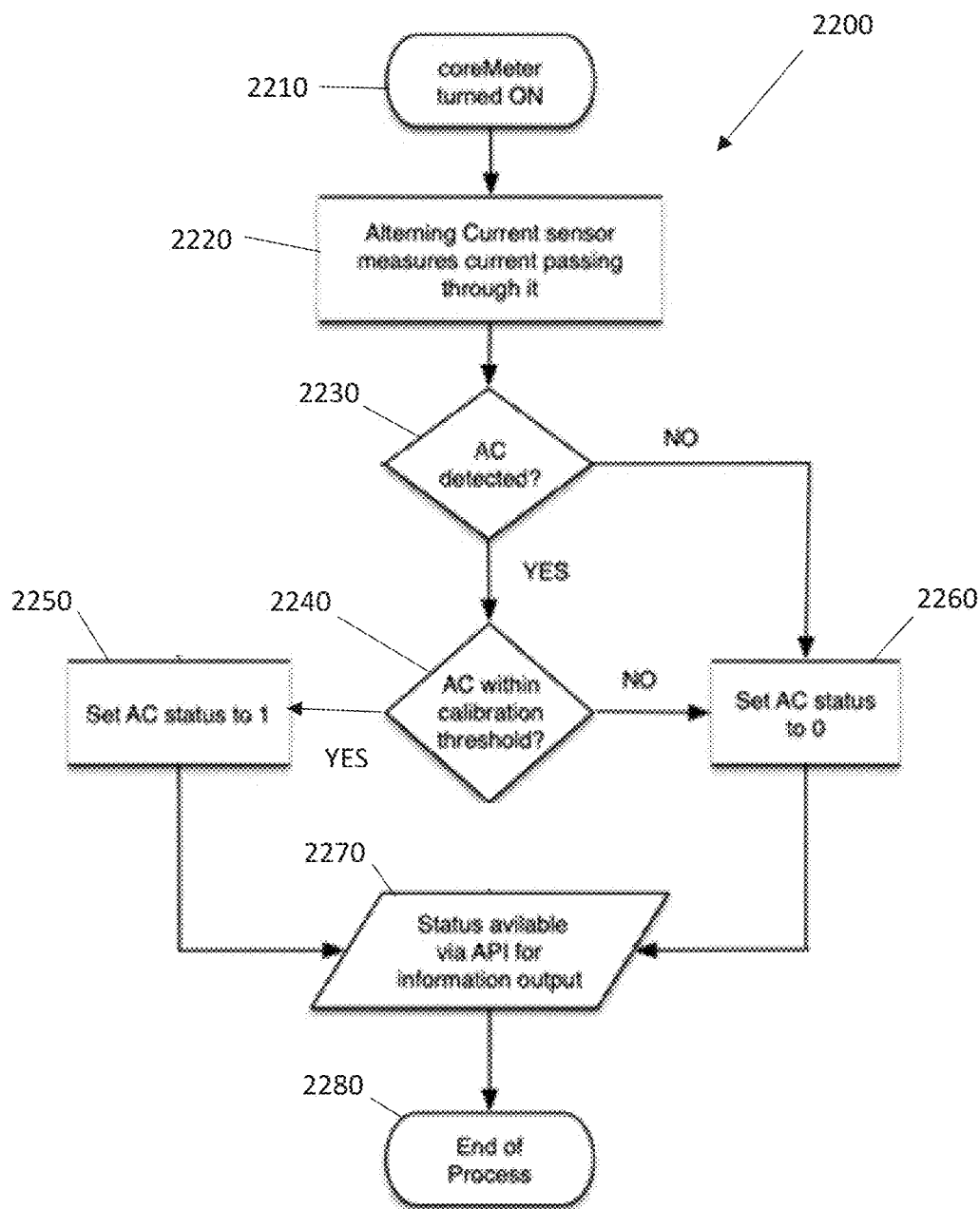


FIG. 23



FIG. 24

U.S. Patent

Feb. 23, 2021

Sheet 43 of 43

US 10,932,002 B2



FIG. 25

US 10,932,002 B2

1

**CROSS-MEDIA MEASUREMENT DEVICE
AND METHOD****CROSS-REFERENCE TO RELATED
APPLICATIONS**

The present application claims priority from U.S. Provisional Patent Application Ser. No. 62/871,789, filed Jul. 9, 2019, the entire contents of which are incorporated herein by reference.

FIELD

The present disclosure relates to the field of electronic media measurement, and particularly devices and methods for determining audience measurement for numerous media events.

BACKGROUND

Media content has been an important part of modern life for well over a century. Media content consumed in homes includes various sources, including cable television, over-the-air (OTA) television, recorded video (e.g., DVD), gaming consoles, and various internet sources offering media content via a high speed internet connection (i.e., over-the-top (OTT) content). While consumers enjoy the ability to watch this media content at will, media providers and advertisers have a vested interest in knowing exactly what media content is actually being consumed (i.e., media that is actually viewed, watched, or otherwise on a screen). By knowing this information, media providers and advertisers are better equipped to create new content and strategically place ads within such content.

The use of statistics offers one convenient approach to measuring media content consumption across a large population. To accomplish this, a group of households are recruited to serve on a "panel" intended to be representative of a larger population (e.g., the individuals in ten homes representative of a neighborhood, the individuals in one hundred homes representative of a city, etc.). Each household includes a number of individual panelists, and each panelist has specific demographic information (e.g., age, sex, ethnicity, income, etc.). By determining what media content that individual panelists are watching, statistical projections can be made about what media content is being consumed by the population as a whole.

For many years, the primary means for measuring media consumption by individuals within a household was the use of diaries. Each panelist was instructed them to keep a physical log of all content that they watched during the week. At the end of every day, each panelist would have a diary log listing everything the panelist watched for the day. This diary approach is still used in local markets to determine what ads to show during local news, etc. While the diary approach is capable of generating valuable information, there are many shortcomings. For example, panelists are notoriously inconsistent on accurately recording what they watched. Panelists often forget to log data, or simply cannot remember all of the media content they watched. The diary approach is also slow to assemble data, as physical diary logs must be collected from each user, the data compiled, and assembled into a user format. These activities not only take a significant amount of time and manpower, but are also subject to human error, making the diary approach to measuring media consumption costly and unreliable.

2

Numerous attempts have been made to implement technological solutions to the measurement of household media consumption in the hopes of addressing the shortcomings of diaries. Inaudible watermarks are an example of one such technological solution that has been attempted in the past. With this approach, audio signals that are inaudible to the human ear are incorporated into media content and captured by listening devices worn by the panelist. The watermark may be, for example, a series of inaudible tones, chimes, or other audio that are periodically played during television programs and/or advertisements. Each panelist is assigned a pager or other listening device that is worn on the panelist while at home. When the watermark is played, it is inaudible to the panelist, but the pager assigned to the panelist records the watermark, and a determination is made that the panelist was watching the media content at the time the watermark was played. Of course, the watermark approach also has numerous shortcomings. For example, media content is often not associated to a panelist because the panelist forgets to wear their pager, or the pager loses power. Also, media content is often incorrectly associated with a panelist because the panelist removes the pager from his or her person and subsequently leave the room. As a result, in accurate data is often collected by the pagers. Moreover, with the watermark approach, only content that includes a watermark is capable of being captured. Many types of media that a user enjoys may not have a watermark (e.g., non-participating programs and advertisements, DVDs and other recorded media, gaming systems, etc.). As a result, panelists may watch a significant amount of media that is not captured in any way by the system. Therefore, while watermark systems offer some benefits over the conventional diary approach, many deficiencies remain in these systems.

Another example of a past technological solution to measuring household media consumption is fingerprinting using various techniques such as audio or video "automatic content recognition" (ACR). Audio ACR involves recording all of the audio aired (either OTA or on cable TV) on various channels in order to provide a library of audio data within a database. The recorded audio include both programming and advertising. Data associated with each audio signal is also catalogued and saved to a programming grid that identifies all of the content played at different times on different channels. For example, for a particular program, information about the channel that aired the program, the time of airing, the actors in the program, title and other episode data may be catalogued and saved. Thereafter, audio ACR involves periodically recording a short clip of the audio signal (e.g., ten seconds) that is output from the television of the panelist. After an audio fingerprint is captured at a particular time, it is sent to the cloud and compared to each of the audio recordings in the library. When a match is found, i.e., when the fingerprint is matched to a particular portion of one of the audio recordings in the library, the media content is identified based on the grid. In this manner, audio ACR is capable of determining what aired content a user watched at any given time. However, because of difficulties with accurately collecting fingerprints for both programming and advertising, these libraries are generally separate, and either programming fingerprints or advertising fingerprints are captured for a single household, but not both.

Like other prior art solutions to measurement of household media consumption, audio ACR has numerous shortcomings. First, it will be quickly recognized that the computer resources required for audio ACR are enormous, including both the memory and processing power required to store massive amounts of audio content in the library and

US 10,932,002 B2

3

subsequently compare each captured audio fingerprint to the audio content in the library. Similarly, the time required for system processors to actually compare each fingerprint to all audio recordings in the library is also significant. Moreover, the time and resources required to actually create the programming grid are also enormous. Audio ACR also has other shortcomings. For example, the audio signal captured by audio ACR is often noisy (e.g., because of noisy households), and incapable of recognition. Also, determining who was watching the identified content is problematic because users are required to actively register in association with the content (e.g., by pressing a button on the audio ACR device). Furthermore, audio ACR is only capable of identifying content that is associated with aired programs (i.e., either on cable or OTA). Audio ACR is incapable of identifying media content associated with gaming or OTT content. Audio ACR is also incapable of determining both programming and advertising consumption for a single household of panelists. Therefore, while audio ACR is capable of providing some advantages, it has numerous shortcomings that do not address the current needs in the industry.

Video ACR (also known as pixel ACR) is another form of ACR that operates similar to audio ACR, but monitors a video fingerprint instead of an audio fingerprint. In particular, video ACR records a number of pixels as a particular location on the screen, and then compares the recorded pixels to a library of pixels associated with programming and advertising content. While video ACR solves a few of the problems of audio ACR, such as noise associated with the audio signal, video ACR has related shortcomings. For example, video ACR is only capable of monitoring aired content associated with a particular time and channel. Video ACR is not capable of identifying content provided from other sources such as gaming consoles, video players (e.g., DVD players), or OTT content.

In view of the foregoing, it will be recognized that consumer's media viewing habits have far outpaced current measurement technology solutions which are unable to truly capture an audience's media exposure. The media consumption measurement industry is relying on multiple measurement sources for each device and relies on modeling to infer the measurement gaps. It would be advantageous to provide a system for media content measurement that is robust, capable of identifying content from all media sources within a household, and is not burdened by the shortcomings of past devices and methods for collecting and identifying consumed media content.

SUMMARY

In accordance with one exemplary embodiment of the disclosure, there is provided a method of identifying media content presented on a display device. The display device includes a screen and a speaker, and is in communication with a content gateway. The media content presented on the display device is provided by a video signal comprising a series of frames. The method comprises determining, at a processor within the gateway, a selected input source providing the video signal, wherein the selected input source is one of a plurality of input sources including at least a first input source and a second input source. The method further comprises selecting a first set of content identification rules when it is determined that the selected input source is the first input source, wherein the first set of content identification rules define a first trigger event and a first algorithm for analyzing one or more of the frames of the video signal following the first trigger event. Furthermore, the method

4

comprises selecting a second set of content identification rules when it is determined that the selected input source is the second input source, wherein the second set of content identification rules define a second trigger event and a second algorithm for analyzing one or more frames of the video signal following the second trigger event, and wherein the second set of content identification rules is different from the first set of content identification rules. Additionally, the method comprises applying the selected first set or second set of content identification rules to the video signal in order to generate content identification data for the media content presented on the display device, wherein applying the selected first set of content identification rules includes waiting for the first trigger event and applying the first algorithm to one or more frames of the video signal following the first trigger event, and wherein applying the selected second set of content identification rules includes waiting for the second trigger event and applying the second algorithm to one or more frames of the video signal following the second trigger event.

In accordance with another exemplary embodiment of the disclosure, a non-transitory computer-readable medium is disclosed for identifying media content provided by a video signal delivered to and presented on a display device. The computer-readable medium includes a plurality of instructions stored thereon that, when executed by a processor, cause the processor to determine a selected input source providing the video signal, wherein the selected input source is one of a plurality of input sources including at least a first input source and a second input source. The instructions further cause the processor to select a first set of content identification rules when it is determined that the selected input source is the first input source, wherein the first set of content identification rules define a first trigger event and a first algorithm for analyzing one or more frames of the video signal following the first trigger event. The instructions also cause the processor to select a second set of content identification rules when it is determined that the selected input source is the second input source, wherein the second set of content identification rules define a second trigger event and a second algorithm for analyzing one or more frames of the video signal following the second trigger event, the second set of content identification rules being different from the first set of content identification rules. Additionally, the instructions cause the processor to apply the selected first set or second set of content identification rules to the video signal in order to generate content identification data for the media content presented on the display device, wherein application of the selected first set of content identification rules causes the processor to wait for the first trigger event and apply the first algorithm to one or more frames of the video signal following the first trigger event, and wherein application of the selected second set of content identification rules causes the processor to wait for the second trigger event and applying the second algorithm to one or more frames of the video signal following the second trigger event.

In accordance with yet another exemplary embodiment of the disclosure there is presented a gateway for identifying media content presented on a display device including a screen and a speaker. The gateway includes a plurality of input ports, an output port, and a processor. The plurality of input ports include at least a first input port and a second input port. The output port is configured to transfer a video signal received at the first input port or the second input port to the display device, wherein the video signal includes a series of frames that provide the media content. The pro-

US 10,932,002 B2

5

cessor is configured to execute a computer application comprising a plurality of instructions which are configured to, when executed, cause the gateway to determine a selected input port providing the video signal, and select a first set of content identification rules when it is determined that the selected input port is the first input port, wherein the first set of content identification rules define a first trigger event and a first algorithm for analyzing one or more frames of the video signal following the first trigger event. The instructions further cause the gateway to select a second set of content identification rules when it is determined that the selected input port is the second input port, wherein the second set of content identification rules define a second trigger event and a second algorithm for analyzing one or more frames of the video signal following the second trigger event, and wherein the second set of content identification rules is different from the first set of content identification rules. Additionally, the instructions cause the gateway to apply the selected first set or second set of content identification rules to the video signal in order to generate content identification data for the media content presented on the display device, wherein application of the selected first set of content identification rules causes the processor to wait for the first trigger event and apply the first algorithm to one or more frames of the video signal following the first trigger event, and wherein application of the selected second set of content identification rules causes the processor to wait for the second trigger event and apply the second algorithm to one or more frames of the video signal following the second trigger event.

The above described features and advantages, as well as others, will become more readily apparent to those of ordinary skill in the art by reference to the following detailed description and accompanying drawings. While it would be desirable to provide a method and system for media measurement that provides one or more of these or other advantageous features as may be apparent to those reviewing this disclosure, the teachings disclosed herein extend to those embodiments which fall within the scope of any eventually appended claims, regardless of whether they include or accomplish one or more of the advantages or features mentioned herein.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows a diagram of a system for cross-media content measurement including a media content gateway positioned within a household and in communication with a remote server;

FIG. 2A shows a block diagram of the gateway of FIG. 1;

FIG. 2B shows a schematic arrangement for the electronic components within the gateway of FIG. 1;

FIG. 2C shows a perspective view of one embodiment of a housing for the gateway of FIG. 1;

FIG. 2D shows a perspective view of an alternative embodiment of a housing for the gateway of FIG. 1;

FIG. 3 shows a block diagram of the remote server of FIG. 1;

FIG. 4 shows inclusion of the gateway of FIG. 1 in an entertainment center of a household along with various other media components;

FIG. 5A shows a first flowchart of general operation of the gateway of FIG. 1;

FIG. 5B shows a second flowchart of general operation of the gateway of FIG. 1;

6

FIG. 5B1 shows a flowchart of implementation of a first set of content identification rules when a selected input source is OTA television;

FIG. 5B2 shows FIG. 5B1 shows a flowchart of implementation of a second set of content identification rules when a selected input source is a set-top box;

FIG. 5B3 shows a flowchart of implementation of a third set of content identification rules when a selected input source is OTT content;

FIG. 5B4 shows a flowchart of implementation of a fourth set of content identification rules when a selected input source is a disc player or a video game console;

FIG. 6 shows an illustration of content identified via the gateway of FIG. 1 from a household of exemplary panelists;

FIG. 7 illustrates a dataset of media consumption data for a panel collected at the remote server of FIG. 1;

FIG. 8 is a diagram showing exemplary inputs into the gateway of FIG. 1 and an associated HDMI output to the television;

FIG. 9 shows an example of media content presented on a television and metadata extracted from a frame of the content;

FIG. 10A illustrates an infrared trigger event used in association with a set of content identification rules in the gateway of FIG. 1;

FIG. 10B illustrates a time based trigger event used in association with the gateway of FIG. 1;

FIG. 10C illustrates several content banners displayed on the television of FIG. 1;

FIG. 10D is an exemplary content log illustrating use of the content banner of FIG. 10C as a trigger event;

FIG. 10E illustrates several content mosaics displayed on the television of FIG. 1;

FIG. 10F is an exemplary content log illustrating use of the content mosaic of FIG. 10E as a trigger event;

FIG. 10G is an exemplary content log illustrating a user's path through one of the content mosaics of FIG. 10E;

FIG. 10H illustrates a network logo displayed on the television of FIG. 1;

FIG. 10I is an exemplary content log illustrating use of the network logo of FIG. 10H as a trigger event;

FIG. 10J illustrates a trigger event provided by a scene change within a series of video frames;

FIG. 10K illustrates a brand recognition trigger event used in association with the gateway of FIG. 1;

FIG. 11 is a plan view of an exemplary remote control for the gateway of FIG. 1;

FIG. 12 is a table of Wi-Fi handshake information collected at the gateway of FIG. 1;

FIG. 13 is a flowchart of a method for registering and de-registering panelists at the gateway based on a Wi-Fi signal strength detected at the gateway of FIG. 1;

FIG. 14 shows a perspective view of exemplary wearable electronic devices for use with panelist registration to the gateway of FIG. 1;

FIG. 15 is a flowchart of a method for registering and de-registering panelists at the gateway based on a Bluetooth communications with a wearable electronic device at the gateway of FIG. 1;

FIG. 16 is an exemplary television with media content presented on the screen and registered panelists displayed as avatars on the screen according to the method of FIG. 15;

FIG. 17 shows the exemplary television of FIG. 16 when the avatars are muted on the screen;

FIG. 18 shows an exemplary network traffic log when the gateway of FIG. 1 operates in a router mode.

US 10,932,002 B2

7

FIG. 19 is a flowchart of a method for detecting network traffic when the gateway of FIG. 1 operates in the router mode;

FIG. 20 is a flowchart of a method for detecting network traffic when the gateway of FIG. 1 operates in a promiscuous mode;

FIG. 21A is a tree diagram of exemplary data packages generated by the gateway of FIG. 1 based on different rule sets for different input sources;

FIG. 21B illustrates a database having exemplary data associated with the data packages of FIG. 21A;

FIG. 21C illustrates additional data within the database of FIG. 21B;

FIG. 22 is a flowchart of a method of determining television on/off state using HDMI-CEC detection at the HDMI output port of the gateway of FIG. 1;

FIG. 23 is a flowchart of a method of determining television on/off state using the AC power detection circuit in the gateway of FIG. 1;

FIG. 24 is a first exemplary embodiment of a graphical user interface for the gateway of FIG. 1; and

FIG. 25 is a second exemplary embodiment of a graphical user interface for the gateway of FIG. 1.

DESCRIPTION

A system and method for cross-media content measurement is disclosed herein. As shown in FIG. 1, the system for cross-media measurement 100 includes a media gateway 110 connected to a television 200 or other display device within a household. The gateway 110 is connected to various media sources within a household 202, including both wired media sources 210 and wireless media devices 220. The gateway 110 is configured to detect consumption of and identify media content presented on both the television 200 and the various wireless devices 220. The gateway 110 is further configured to associate one or more panelists 204 with the identified media content. Data collected by the gateway 110 is transmitted to a remote server 310 via the internet 290 or other wide area network. The remote server 310 may perform additional processing on the data collected by the gateway 110 in order to determine the specific media content consumed by each of the specific panelists.

Gateway Architecture

FIGS. 2A-2C show an exemplary embodiment of the media gateway 110 (which may also be referred to as a "content recognition meter" or "Coremeter"). FIG. 2A shows a block diagram of the gateway 110. Similarly, FIG. 2B shows a schematic layout for the gateway 110. FIG. 2C shows an exemplary housing 112 for the gateway. It will be appreciated that the embodiment of the media gateway 110 shown in FIGS. 2A-2C is only one exemplary embodiment of a media gateway. As such, the exemplary embodiment of the media gateway 110 of FIGS. 2A-2C is merely representative of any of various manners or configurations of the media gateway 110 or other data processing systems that are operative in the manner set forth herein.

The media gateway 110 is provided in a housing 112, cabinet or the like, and includes a number of ports and associated electronic components enclosed within the housing 112. As can be seen in FIG. 2C, the housing 112 is a simple cube-shaped box structure with a solid color provided on the base (e.g., a gray color), and a contrasting color provided on an upper portion of the housing above the base (e.g., a black color). The display 152 of the gateway 110 is visible on the upper portion of the housing. The simple design of the housing 112 is intended to be both rugged and

8

aesthetically pleasing. However, FIG. 2C is only one of numerous possible embodiments for the housing 112. FIG. 2D shows an alternative embodiment of the housing that does not include the display 152, but includes a number of additional antennas 128 associated with the wireless transceiver 124.

With particular reference now to FIG. 2A, the electronic components of the media gateway 110 include processing circuitry/logic 114, a memory 116, a communications module 120, an infrared receiver 122, a wireless transceiver 124, a cellular transceiver 126, a number of input/output ports 130, a power module 150, a display 152, a microphone 154, and a speaker 156.

The processing circuitry/logic 114 is operative, configured and/or adapted to operate the content gateway 110 including the features, functionality, characteristics and/or the like as described herein. To this end, the processing circuitry/logic 114 is operably connected to the memory 116, and various other components including the communications module 120, the I/O ports 130, the power module 150, the display 152, the microphone 154, and the speaker 156. The processing circuitry 114 may be provided by one or more commercially available microprocessors, such as a quad core 1.8 GHz or faster processor, such as those sold by Intel Corporation or AMD, Inc. The processing circuitry 114 may be included on a single board/processor, or may be split amongst a number of different boards and processors within the gateway 110. For example, in at least some embodiments, the processing circuitry includes a CPU, a motherboard, and one or more additional processing modules, such as a video capture module 115 (see FIG. 2B).

The memory 116 may be of any type of device capable of storing information accessible by the processor, such as solid state memory, hard drives, memory cards, ROM, RAM, write-capable memories, read-only memories, discs, flash memory, or any of various other computer-readable medium serving as data storage devices as will be recognized by those of ordinary skill in the art. In the embodiment, shown in FIG. 2B, the memory includes 4 GB (or more) DDR3 RAM as well as 8 GB (or more) SSD storage.

The memory 116 is configured to store both instructions 160 for execution by the processing circuitry/logic 114, as well as data 170 for use by at least the processing circuitry/logic when running one or more of the programs/software engines included in the instructions 160. In the embodiment described herein, the instructions 160 include various software programs/engines, including OTT apps 161, a content capture engine 162, a client-side content identification engine 164, a panelist registration engine 166, a content overlay engine 168, as well as numerous other computer programs. It will be recognized that the instructions 160 also include various additional programs that are not discussed in detail herein. For example, the instructions 160 include a hardware interface application programming interface (API) that allows the gateway 110 to interact with various hardware components such as the communications module 120 and associated transceivers 122, 124, 126, I/O ports 130, power module 150, display 152, etc.

The OTT apps 161 include any of various apps available to or downloaded by the user for use via the gateway 110. The OTT apps may 161 include any of various applications for streaming OTT content, such as the Amazon Prime app, Hulu, Netflix, etc.

The content capture engine 162 is configured to select, copy and save certain screenshots, video snippets, and/or audio associated with the selected video source that is delivered to the television 200. The content captured may be

from any of various media sources, including one of the sources connected to one of the I/O ports **130** (e.g., one of the HDMI-IN ports **132**) as well as any additional sources such as content provided by one of the OTT apps **161** included in the memory **116** of the gateway **110**. The content capture engine **162** captures/copies specific types of content (e.g., video frames) at certain times. The capture of content may occur periodically and/or be determined based on instructions from the content identification engine **164**. In at least some embodiments, the content capture engine **162** interacts with the hardware interface API and captures content every time an infrared signal is received from the remote control **206**. Content captured with the content capture engine **162** may be processed by the content identification engine **164** and/or transmitted to the remote server **310** for additional processing.

The client-side content identification engine **164** is configured to monitor, analyze and identify content presented on the television screen and other display devices using a multi-layered approach to content identification. As explained in further detail below, the multi-layered approach involves the application of different rules to content review and identification based on the source of the content. The content identification engine **164** advantageously uses machine learning to improve upon content identification over time. The content identification engine performs a first round of content recognition on the captured media by using machine learning models. If the client-side content identification engine **164** is able to detect the required information with enough accuracy, the detected information (e.g., channel and other metadata) are transmitted directly to the cloud and stored in a database of the remote server together with panelist registration information. On the other hand, if the content is not identified with an acceptable level of accuracy, the content is sent to the remote server **310** for further processing by more robust machine learning engines.

The panelist registration engine **166** operates in association with the content identification engine **164** to identify certain panelists for association with identified content. The panelist registration engine **166** works directly with the hardware interface API in order to detect the household member's presence using any of various means. For example, the panelist registration engine **166** may analyze the power of the Wi-Fi signal that arrives at the gateway **110** from wireless devices **220** (e.g., smartphones and other mobile devices) associated with each of the panelists. As another example, the panelist registration engine **166** may interact with a Bluetooth chipset **123** on the wireless transceiver **124** in order to detect the presence of Bluetooth wearables that are assigned to one or more of the panelists (typically child panelists). It will be recognized that the terms "register" and "registration" as used herein with respect to one or more panelists refers to the condition of a panelist being in proximity to the television or other display device such that the panelist is associated with identified content presented on the display device; the terms "register" and "registration" as used herein with respect to one or more panelists does not refer to such panelists being users of the system and/or simply having demographic information for the panelist saved in the system **100**.

The content overlay engine **168** is configured to display certain content on the television in association with media presented thereon. For example, as described in further detail below, the content overlay engine **168** is configured to overlay avatars representing the currently registered panelists over the video content presented on the television **200**. The content overlay engine **168** is also configured to provide

various graphical user interfaces (GUIs) for use in association with gateway operation. In at least some embodiments, the content overlay engine **168** also includes the software that operates the video capture module **115** of the gateway **110**. In this embodiment, the content overlay engine **168** determines the video signal output at the HDMI-OUT port **133** of the gateway. Accordingly, the content overlay engine **168** may be configured to determine the programming/media content presented on the television in addition to any overlays on the programming/media content.

With continued reference to FIG. 2A, the data **170** stored in the memory **116** includes panelist data **172**, a viewing log **174**, training data **176**, and saved content. The panelist data **172** includes data related to all panelists **204** in the household **202**. The panelist data **172** may include any of various types of demographic data such as age, sex, income level, etc. for each panelist within the household **202**. In at least some embodiments, the panelist data **172** also includes personalized information collected about the panelist at the time of registration. For example, the panelist data **172** may include education level, type of smartphone owned, type of automobile owned, pet ownership, vacation preferences, sports preferences, food preferences, etc.

The viewing log **174**, includes information about the content viewed by the individual panelists. For example, the viewing log **174** may include individual data indicating that a particular panelist was watching a particular program at a particular time. For example, the viewing log **174** may include data indicating two panelists **204** from the household **202** were watching "Game of Thrones" on HBO via a cable box at 10 pm on Jul. 9, 2020. The viewing log **174** may be saved to the memory **116** of the gateway **110** for some period of time, and then periodically transmitted to the remote server **310** (e.g., at the end of every day), and/or erased after some period of time (e.g., after one month).

The training data **176** includes data that is used by the machine learning features of the client-side content identification engine **164**. The training data includes a number of exemplary video frames for different input sources, and the appropriate content identification for such video frames. The training data **176** is used to train the content identification engine **164** to appropriately identify content from a video frame. The training data **176** is periodically updated (e.g., daily, weekly, etc.) by uploading additional training data from the remote server **310**. This updated training data allows the content identification engine **164** to experience incremental learning, thus allowing the content identification engine to more reliably identify content from any of various input sources.

The saved content **178** includes frames of captured video that are saved for future reference. While much of the content identification process occurs locally on the gateway **110**, in certain situation frames that require further processing are temporarily stored in the saved content **178**. These frames may then be transmitted to the remote server **310** for further processing. For example, when certain types of frames that require more in-depth processing (e.g., face identification) are identified, these frames are temporarily stored with the saved content **178**, and then subsequently transferred to the remote server **310** for further processing. In some instances, transfer of the saved content occurs along with content data that was identified at the gateway **110** (e.g., text data associated with each frame).

In view of the foregoing, it will be recognized that the data **170** is used by the computer programs **162**, **164**, **166**, **168** utilize the data **170** in order to provide the functionality of the cross-media content identification system **100** described

US 10,932,002 B2

11

herein. A computer program product implementing an embodiment disclosed herein, including any of the above-mentioned programs may comprise one or more computer-readable storage media storing computer instructions executable by a processor to provide an embodiment of a system or perform an embodiment of a method disclosed herein. Computer instructions (e.g., the client-side content identification engine **164**) may be provided by lines of code in any of various languages as will be recognized by those of ordinary skill in the art. A “non-transitory computer-readable medium” may be any type of data or storage medium that may store computer instructions, including, but not limited to a memory card, ROM, RAM, write-capable memories, read-only memories, hard drives, discs, flash memory, or any of various other computer-readable medium.

With continued reference to FIG. 2A, the communication module **120** of the gateway **110** provides an interface that allows for various types of communication with any of various media devices. The communications module **120** is specifically configured for both wired and wireless communications with various media devices and other electronic devices. The communications module **120** is configured for wireless communications via the I/O ports **130**, and is configured for wireless communications via various wireless interfaces, including an infrared receiver **122**, a wireless transceiver **124**, and a cellular transceiver **126**.

The communications module **120** connects the gateway **110** to the household's internet service provided by an internet service provider (e.g., via cable or fiber delivered to a household modem). The connection to the internet may be by wired communication (e.g., over the Ethernet port **142**) or wireless communication (e.g., over the wireless transceiver **124**). For example, the wireless transceiver **124** connected to the communications module **120** specifically includes a Wi-Fi chipset **125**, thus allowing the communications module **120** to communicate with an existing Wi-Fi network provided by an internet service provider. Connection of the communications module **120** to the internet allows the gateway **110** to serve as a router in a new wireless network within the household. Thus, the gateway **110** serves as a Wi-Fi access point for all wireless network devices **220** within the household, including both mobile devices (e.g., smartphones and tablets) and stationary devices (e.g., desktop computers and the television **200**). Besides acting as a router/Wi-Fi access point, the Wi-Fi chipset **125** also allows the gateway **110** to sniff the traffic on each mobile device and detect mobile browsing history, searched keywords and target URLs. Additionally, the Wi-Fi chipset **125** detects the signal strength (e.g., RSSI) of each mobile device at the gateway, thus allowing the gateway **110** to detect proximity of a mobile device (and the associated user) to the gateway. Thus, it will be recognized that the Wi-Fi chipset **125** provides for triple functionality: (i) it can connect to a Wi-Fi network as a client (e.g., in order to transmit captured data to a central server or receive software updates); (ii) it can act as a Wi-Fi access point such that other Wi-Fi devices connect to it (e.g., in order to perform network sniffing functions to determine the kind of contents the user is consuming); and (iii) it can act as a Wi-Fi beacon in order to detect the proximity of a mobile device (e.g., in order to allow for passive user presence detection in the same room as the television).

In addition to the wireless transceiver **124** and associated Wi-Fi chipset **125**, the communications module **120** further includes a cellular transceiver **126** (or other wide area network transceiver) and associated chipset. The cellular

12

transceiver **126** may include a cellular modem that facilitates internet communications between the gateway **110** and any of various remote computers via the cellular telephony network (e.g., 3G/4G/5G/LTE networks within the American frequency specification). In this manner, the gateway **110** is equipped with redundant functionality that allows for internet communications via any one of various available network connections, including: (i) an Ethernet connection, (ii) a Wi-Fi connection, or (iii) the cellular telephony network connection.

In addition to the internet connection capabilities, including Wi-Fi and cellular capabilities as discussed above, the communications module **120** also includes further wireless communications capabilities. For example, the infrared receiver **122** allows the communications module **120** to receive infrared signals from a remote control or other infrared-equipped device. Additionally, the wireless transceiver **124** may also provide other communications capabilities using any of various known hardware, software and related communications protocols. For example, the wireless transceiver **124** is also configured to provide short-range wireless communications (e.g., via the low emissions Bluetooth chipset **123**) with any of various short-range communications devices. The short-range wireless communications provides additional functionality for the gateway **110**, such as additional remote control functionality, or panelist registration functionality as will be explained in further detail below.

The I/O ports **130** include a number of ports that are accessible through the housing **112** of the gateway **110**. As best shown in FIG. 2B, the I/O ports **130** include a plurality of HDMI-IN ports **132** (e.g., 4-8 ports), an HDMI-OUT port **133**, a plurality of USB ports **134** (e.g., 2-4 ports), at least one CVBS-IN port **136**, a CVBS-OUT port **137**, and a digital antenna in port **138**. The HDMI-IN ports **132** are configured to connect any of a plurality of different wired media sources **210** to the gateway **110** using an HDMI cable having an HDMI connector. Typical media sources that may be connected to the gateway **110** via the HDMI-IN ports **132** include cable boxes, Blu-ray and DVD players, OTT streaming devices (e.g., Apple TV, Roku, Amazon Firestick, etc.), video game consoles (e.g., Sony PlayStation, Microsoft X-Box, Nintendo Switch, etc.), video cameras, and any number of other media devices.

Similar to the HDMI ports, the USB ports **134** are configured to connect any of a plurality of different wired media sources **210** to the gateway **110** using a USB cable with a USB connector. Typical media sources that may be connected to the gateway **110** via the USB ports **134** include video cameras and computer devices such as tablets, laptops, and desktop computers. The USB ports **134** are particularly equipped to allow system administrator to perform activities such as operating system updates, media files transfer, extend storage capacity, add external dongles of any kind to expand the hardware capacity, and connect peripheral accessories such as biometric readers, webcams or other sensors.

The at least one CVBS port **136** is available in the event that a wired media source **210** does not include an HDMI port (e.g., an older media device, such as a VCR), in the event that composite video cables are available but an additional HDMI cable is not available to the user during set-up of the gateway **110**, or in the event that all of the HDMI ports **132** are in use. The digital antenna in port **138** is generally a coax connection port that receives input from an HDTV antenna. The digital antenna in port is connected

US 10,932,002 B2

13

to a digital ATSC TV tuner **139** that allows the user to receive OTA content from any of a number of local TV providers.

Each of the video input ports, including the HDMI-IN ports **132**, USB ports **134**, CVBS-IN port **136**, and TV tuner **139** are connected to the video capture module **115**. The video capture module **115** is configured to receive video signal inputs from the various ports **132**, **134**, **136**, **139**, and act as a switch to select one of the inputs to be output to the television via the HDMI-OUT port **133** (or alternatively, the CVBS-OUT port **137**). Selection of the appropriate video signal for output via the HDMI-OUT port **133** is typically determined by user/panelist selection of one of the input ports via the remote control or other means, thus indicating the user's preferred viewing source. The video signal from the selected input port is then output to the television for presentation to the user. Accordingly, a single HDMI input is received at the television **200** from the gateway **110**, and there is no need for the user to switch video inputs at the television. Instead, selection of video inputs occurs at the gateway **110**. Furthermore, because the video signal to the television is directed through the gateway **110**, the content overlay engine **168** is configured to overlay additional content, such as legends, alerts and registered persons, on the television screen via the HDMI-OUT port **133** to the television **200**.

In addition to the video input ports, the I/O ports also include several additional ports including an SD card expansion slot **140**, an Ethernet port **142**, and an AC output port **144**. The SD card expansion slot **140** allows the user to insert an SD card so that content thereon can be read by the gateway **110**. For example, the SD card expansion slot **140** may be used as an expansion slot for additional storage capabilities when connectivity problems exist with the gateway **110** (e.g. due to faulty Wi-Fi at the household or problems with the cellular network). Alternatively, the SD card expansion slot **140** may be used to provide updates or other information for use by the processing circuitry **114**. The Ethernet port **142** (e.g., an RJ45 10/100 MBPS Ethernet port) is configured to connect to the household modem provided by an internet service provider (ISP). The Ethernet port **142** is typically used when the internet modem provided by the ISP is in close proximity to the gateway **110**. If this modem is not in close proximity to the gateway **110** (or otherwise available for wired connection), the gateway utilizes the wireless transceiver **124** or the cellular antenna **126** to connect to the household router provided by the ISP. The AC output port **144** is connected to the power module **150**. As explained in further detail below, the AC output port **144** is configured to receive the power cord from the television **200** and provide AC power to the television.

The power module **150** is adapted to provide power to both the gateway **110** as well as the television. To this end, the power module **150** includes an internal power supply that is configured to plug in to an AC power outlet within the household **202**. The power module **150** is also connected to an internal battery **148**. The power module charges the internal battery **148**, and in turn, receives power from the internal battery **148** in the event power from the household AC power outlet is not delivered to the power module **150**. As shown in FIG. 2B, the internal power supply is connected to the AC output port **144** and provides power to the AC output port. In order to facilitate delivery of AC power, the AC output port **144** includes an AC receptacle **145** that other electronic devices may be plugged into. When the television **200** is plugged into the receptacle **145** of the AC output port **144**, the television receives power via the gateway **110**.

14

The power module **150** also includes a TV ON/OFF detection circuit **151**. The TV ON/OFF detection circuit **151** is configured to determine whether the TV connected to the gateway **110** is on or off in one of two different ways. First, if the TV is plugged into the AC power port **144**, the circuit **151** detects the amount of power flowing to the television (e.g., via an AC loop sensor or other current sensor or via a shunt resistor or other voltage sensor). When the amount of power provided to the television **200** is less than a threshold amount (i.e., indicating that the screen is not illuminated), the television is determined to be powered off. When the amount of power provided to the television is greater than a threshold amount (i.e., indicating that the screen is illuminated), the television is determined to be powered on. Second, if an HDMI cable connects the gateway **110** to the television, the HDMI CEC (consumer electronics control) feature may be used to detect whether the television **200** is powered on or off. As explained in further detail below, determining whether the television **200** is powered on or off may be used to (i) establish the measurement on/off times based on the times the user is actually watching television, (ii) optimize resource usage by processing and transmitting information only when the user is watching television, and (iii) detect user presence only during television viewing times. Also, because the gateway can detect the ON/OFF television state, the power module **150** is further configured to automatically turn on (full power) when the television **200** is turned on, and automatically turn off (reduced power) when the television is turned off.

With continued reference to FIGS. 2A-2C, the gateway **110** further includes additional electronic components such as a display **152**, a microphone **154**, and a speaker **156**. The display **152** may be a conventional LCD display (e.g., a 16x10 cm LCD display), as shown in FIG. 2C. The display **152** provides simple information for the user such as the current time, date, selected media source (e.g., HDMI **2**), and registered panelists (e.g., 1, 3, 4). The display **152** may also be used by a technician to provide information during diagnostic testing and repair of the gateway **110**. However, the display **152** is not equipped to present video content from any of the sources connected to the ports **130**. In other words, the gateway itself is not capable of acting as a television for user viewing of video signals delivered thereto.

The microphone **154** may be any of various commercially available microphones that are commonly used with electronic devices. The microphone **154** allows the user to provide verbal instructions in lieu of instructions from a remote control or other source (e.g., "Alexa, tune to Netflix," or "Hey Google, what channel is playing the football game"). The speaker **156** is capable of providing audible cues, alerts, reminders, or audio instructions for the user. For example, the speaker **156** may sound an error tone, or may be used to respond to a verbal command from the user (e.g., "The football game is on NBC, channel 12"). In at least one embodiment, the microphone **154** is used to provide additional content identification capabilities, such as audio ACR.

Remote Server

With reference again to FIG. 1, the remote server **310** is positioned at a location that is removed from the household **202** where the gateway **110** resides. The remote server **310** is configured to communicate with the gateway **110** via the internet **290**. Accordingly, both data and instructions may be communicated and shared between the gateway **110** and the remote server **310** via the internet **290**. While only a single remote server **310** is shown in FIG. 1, it will be recognized that this single server **310** is representative of any number of

US 10,932,002 B2

15

remote/cloud servers that may be in communication with the gateway via the internet 290 or other wide area network.

As shown in FIG. 3, the remote server 310 is provided in a housing 312, cabinet or the like, and includes a number of electronic components enclosed therein. In particular, the remote server 310 includes processing circuitry/logic 314, a memory 316, a communications module 320, and a number of input/output ports 330. The communication module 320 of the remote server 310 provides an interface for communication with other devices, and particularly the gateway 110, via the internet. As noted previously, the gateway 110 is connected to the internet using any of various means for establishing internet communications. The remote server 310 may be similarly configured, including configured for wired or wireless connection to the internet. To this end, the I/O ports 330 of the remote server provide the necessary ports, antennas, or other communications hardware required to establish the internet connection.

The processing circuitry/logic 314 of the remote server 310 is operably connected to the memory 316, and various other components including the communications module 320 and the I/O ports 330. Similar to the processing circuitry 114 of the gateway 110, the processing circuitry 314 of the server 310 may be provided by one or more commercially available microprocessors, such as a quad core 1.8 GHz or faster processor, such as those sold by Intel Corporation or AMD, Inc. The memory 316 may also be of any type of device capable of storing information accessible by the processor, such as solid state memory, hard drives, memory cards, ROM, RAM, write-capable memories, read-only memories, discs, flash memory, or any of various other computer-readable medium serving as data storage devices as will be recognized by those of ordinary skill in the art. The memory 316 is configured to store both instructions 360 for execution by the processing circuitry/logic 314, as well as data 370 for use by at least the processing circuitry/logic when running one or more of the programs/software engines included in the instructions 360.

In the embodiment described herein, the instructions 360 include various software programs/engines, including a network-side content identification engine 364. The network-side content identification engine 364 is similar to the client-side identification engine 160, and is configured to identify content presented on a television. However, the network-side content identification engine 364 includes additional functionality and processing capabilities, such as increased machine learning functionality beyond that capable with the client-side content identification engine 160.

The data stored in the memory 316 includes panel-wide data 372, a viewing log 374, and training data 376. The panel-wide data 372 includes data related to all panelists in the system 100, including the panelists 204 associated with the household 202, as well as numerous additional panelists associated with additional households. The panel-wide data 372 may include any of various types of demographic data such as age, sex, income level, etc. for each panelist. The viewing log 374, includes information about the content viewed by panelists, both individually and collectively. For example, the viewing log 374 may include individual data indicating that a particular panelist was watching a particular program at a particular time (e.g., panelist 1 from the household 202 was watching "Game of Thrones" on HBO at 10 pm on Jul. 9, 2020). Additionally, the viewing log 374 may include collective data that indicates that groups of panelists were watching a particular program at a particular time (e.g., 5% of all panelists, or 10% of all males between

16

the ages of forty and fifty were watching "Game of Thrones" on HBO at 10 pm on Jul. 9, 2020). The training data 376 includes data that is used by the machine learning features of the network-side content identification engine 360, as explained in further detail herein.

General Gateway Operation

General operation of the gateway 110 is now be described with reference to FIGS. 4-7. FIG. 4 shows the gateway 110 positioned in a living area of an exemplary household 202. The gateway 110 is positioned in close proximity (e.g., within 5-10 feet) of the television 200. The HDMI-OUT port 133 of the gateway 110 is connected to one of the HDMI-IN ports 201 of the television 200. A plurality of wired media sources 210, including a cable box 212, Blu-ray/DVD player 214, and a gaming console 216 are connected to the gateway 110 using cables connected to the HDMI-IN ports 132 of the gateway 110. Wireless devices 220, including a smartphone 222 and a wearable device 230 (e.g., a smart bracelet), are also connected to the gateway 110 via the wireless transceiver of the gateway 110. A remote control 206 is also configured to communicate with the gateway 110.

The gateway 110 is configured to identify content presented on the television 200 and associate one or more panelists 204 with the identified content. Additionally, the gateway 110 is configured to communicate with the smartphone 222 over a Wi-Fi connection in order to monitor media content presented on the smartphone 222. The gateway 110 is also configured to register panelists 204 and associate registered panelists with identified media content on the television 200. Registration of panelists is accomplished in one of several ways. First, passive registration of panelists may occur using the wireless connection with the smartphone 222 (or other mobile electronic device) to detect user presence in proximity to the gateway 110. Second, passive registration of panelists may occur using the wireless connection, e.g., a Bluetooth connection, with the wearable device 230 in order to detect user presence in proximity to the gateway. Third, active registration of panelists may occur using the remote control 206. To this end, the remote control 206 includes a plurality of dedicated registration buttons for active registration of the panelists to the gateway. Each of the dedicated registration buttons is associated with one of the panelists in the household 202, such that the panelist only needs to press a button in order to actively register their presence in proximity to the television 200. Further detail concerning registration of panelists is provided in further detail below in association with the "Active and Passive Panelist Registration" subheading (and related FIGS. 11-17).

With reference now to FIGS. 5A-5B4, a method 500 of cross-media content measurement is disclosed. The method 500 begins at block 502 when the gateway 110 is turned on (the terms "block" and "step" are used interchangeably herein). At block 504, a check is made that the gateway 110 is receiving power from an AC wall outlet. If the gateway is not receiving power from an AC wall outlet, the device is instructed at block 506 to operate from the battery until power is received from the AC wall outlet. At block 508, the process continues and the gateway goes through the booting process. Once the booting process is complete, the processor is instructed to overlay information about the gateway 110 on the television screen via the HDMI output. At block 512, the gateway detects whether this is the first-time use of the device within a household. If it is a first time use, at block 514 the setup wizard is run, demographic data for each panelist in the household is entered, registration buttons from the remote control are associated to each of the

US 10,932,002 B2

17

panelists, the MAC address of various wireless devices (e.g., smartphones and watches) are associated to the panelists, and wearable devices are associated with panelists (e.g., wearable bracelets for children). Then, at block 516 the gateway checks for an internet connection. If no internet connection is detected, the method continues to block 518 and an alert to request connectivity is issued, and the internet connection wizard is run.

Once an internet connection is established at the gateway 110, the method continues to block 520, and the gateway detects whether power to the television is on (e.g., via a current sensor or a CEC signal from the HDMI connection to the television). If power to the television is not on, no panelist presence or content identification is performed, and at block 522, the gateway 110 periodically sends a diagnostic signal (e.g., every hour) to be sure that the television remains operational. If the television is powered on, the method continues to block 526, and user registration (i.e., user presence in the vicinity of the television) is detected. As noted previously, the gateway 110 is capable of detecting user registration by any one of several means, including active user registration by a pressed button on the remote control 206, passive registration based on signal strength from a user mobile device (e.g., smartphone 222), or passive registration based on a wireless connection to a user wearable device (e.g., bracelet). If no panelists are detected, the method continues to block 528 and an alert is shown on the television instructing any panelists in the room to register (e.g., via the remote control). After one or more panelists are registered by the gateway 110, the method moves on and performs the acts associated with the additional blocks shown in FIG. 5B.

With reference now to FIG. 5B, after one or more panelists are registered with the gateway 110, the method continues at block 530 by determining which input source is selected for presentation on the television (e.g., HDMI 1-3 or OTA), and then obtaining content identification rules for the selected source. Again, the selected input source is the source that video capture module 115 has selected for delivery of the associated video signal to the television. This source may be selected by the user via the remote control, or may simply be the last selected source from a previous television viewing session. As will be explained in further detail below, the selected content identification rules may depend on the specific media input (e.g., cable box, OTA, OTT) as well as the specific device or content provider associated with such box (e.g., Spectrum cable box, Amazon Firestick, Roku, etc.). Depending on the selected input source, the method then implements the associated rules. As shown in FIG. 5B, the method continues at block 532 (and FIG. 5B1) when the selected input source is OTA television. The method continues at block 534 (and FIG. 5B2) when the selected input source is a satellite/cable box. The method continues at block 536 (and FIG. 5B3) when the selected input source is an OTT source (either connected to an input of the gateway 110 or on an app within the gateway). The method continues at block 538 (and FIG. 5B4) when the selected input source is a disc player or game console. While four exemplary methods associated with content identification rules are illustrated in FIGS. 5B1-5B4, it will be recognized that these are merely exemplary methods and numerous additional methods are contemplated for any of various input sources. Accordingly, the particular steps associated with each method, and any associated details (e.g., threshold amounts, times for processing, etc.) are merely illustrative and will change with different sets of content identification rules.

18

With reference now to FIG. 5B1, an exemplary method 540 associated with content identification rules for an OTA television signal is shown. The method 540 includes two different analysis routines 541 and 551 that are processed in parallel. The first routine 541 is a signal analysis routine. This routine 541 begins at step 542 where the tuned signal (from the digital ATSC TV tuner 139) is analyzed. Video signals transferred under the ATSC standard include meta-data that identifies the tuned channel. Accordingly, analysis of the tuned signal with OTA content includes extracting the metadata from the signal in order to identify the content presented on the television screen. At block 543, a determination is made if a threshold period of time (e.g., 5 seconds) has elapsed since the last analysis of the tuned channel. If the period of time has elapsed the method 540 returns to block 542 and the tuned signal is analyzed again. If the threshold period of time has not elapsed, the method 540 proceeds to step 544 where a determination is made whether it is time to transmit the identified content data. If it is not yet time to transmit the data (e.g., once every two seconds) to the remote server 310, the method moves to block 546. However, if it is time to transmit the data, the routine 541 continues, and the identified content data is transmitted at block 545. Then, at block 546, a determination is made whether a change in the selected media source occurred. If a change did not occur, the method returns to block 543 and again analyzes the tuned signal. If a change in the selected media source did occur, the method moves to step 547, where the method returns to step 530 of FIG. 5B and the selected input source for presentation on the television is detected.

With continued reference to FIG. 5B1, in parallel with processing the metadata analysis routine 541, the method 540 associated with the OTA content identification rules also includes a video frame analysis routine 551. The video frame analysis routine 551 begins at block 548 where the processor waits for the next video frame from the OTA video signal. At block 552 a determination is made whether the next frame has been received. If the next frame has not been received, the method moves to step 546 and determines whether there was a change in the selected media source. On the other hand, if the next video frame has been received, the method continues to step 553, and the video frame is captured. Then, at block 554, a machine-learned frame analysis (e.g., logo analysis, face recognition, etc.) is performed on the video frame. At block 555, a determination is made whether the frame indicates a scene change (as explained in further detail below). If there is not scene change at block 555, the method continues to step 556, and a determination is made whether the frame has useful information (i.e., identified additional content data). If additional content data is identified, it is then transferred at block 557. If no useful content data is identified, the method returns to block 548, and waits for a new video frame. On the other hand, if a scene change is detected at block 555, the method continues to block 558 where a scene change timestamp is recorded and/or transmitted. This scene change timestamp is utilized to determine the length of an advertisement. At block 559, the method continues by waiting for a new non-blank screen. The new non-blank screen indicates the beginning of a new content piece (e.g., the start of a commercial, or return to regular programming). When a new non-blank screen is received, the method then returns to block 553 and captures the video frame.

With reference now to FIG. 5B2, an exemplary method 560 associated with content identification rules for a video signal from a satellite/cable box is shown. The method

US 10,932,002 B2

19

begins at step **561** where the processor waits for the next frame in the video signal. At step **562**, a determination is made whether the next frame is received. It will be recognized that the next frame may be an immediately next frame, or could also be a next frame of some predefined number of frames (e.g., one in ten). If the next frame has not been received, the method continues to block **572**, and a determination is made whether there was a change in the selected media source. On the other hand, if the next frame has been received, the method continues to block **563**, and the next frame is analyzed to detect whether a content grid (e.g., banner, guide, mosaic, etc.) or network logo is present within the frame. As explained in further detail below, the algorithms for determining the presence of a content grid or network logo may be considered to be "trigger events." At block **564** the method determines whether the analyzed video frame includes a trigger event in the form of a content grid or network logo. If a content grid or network logo is present, the method continues at step **565**, and a machine-learned content grid analysis is performed and/or machine-learned network logo analysis is performed. The content identification data generated by such analysis is then packaged in a data package and transmitted to the remote server **310**. The method then returns to block **561** where the method waits for the next video frame.

With continued reference to FIG. **5B2**, if no content grid or network logo is detected at step **564**, the method continues to step **567**, and a determination is made whether the analyzed video frame includes another trigger event in the form of a scene change. If a scene change is detected, the method continues to step **568**, and a timestamp of the scene change is generated and stored internally and/or transmitted to the remote server. Thereafter, at step **569**, the method waits for a new non-blank video frame. When a new non-blank video frame is received, this indicates the beginning of a new content piece (e.g., the start of a commercial, or return to regular programming). The method then returns to block **563** and reviews the new frame.

If no content grid or network logo is detected at step **564**, and if no scene change is detected at step **567**, the method **560** continues to step **570**. At step **570**, the method determines whether yet another trigger event has occurred in the form of a predetermined passage of time (e.g., ten seconds) since the last video frame analysis. If the predetermined period has passed and it is time to review another video frame, the method **560** continues to step **571** and a machine-learned algorithm (which may also be referred to herein as a machine-learning module) performs an analysis on the frame. This machine-learned algorithm may be any of a number of machine-learned algorithms configured to detect content from a video frame, such as logo analysis, object detection, face recognition, etc. Following this analysis, the method continues on to step **566**, and any identified content is collected into a data package and transmitted to the remote server **310**.

If a determination is made at step **570** that it is not time to analyze another frame, the process **560** continues to block **572**, where the gateway **110** determines whether there has been a change in the selected media source. If there has been no change in the selected media source, the method **560** returns to step **561** and waits for the next video frame. On the other hand, if there has been a change in the selected media source, the method **560** proceeds to step **573**, where it is instructed to return to step **530** of FIG. **5B**.

With reference now to FIG. **5B3**, an exemplary method associated with content identification rules for a video signal from an OTT provider is shown. The method begins at step

20

581 where the processor waits for the next frame in the video signal. At step **582**, a determination is made whether the next frame was received. If the next frame has not been received, the method continues to block **590**, and a determination is made whether there was a change in the selected media source. On the other hand, if the next frame has been received, the method continues to block **583**, and a machine-learned algorithm analyzes the frame to determine if one of a number of different objects or indicia can be found within the frame. At step **584**, the processor determines whether the frame includes a content mosaic. If the frame does include a mosaic, the process **580** continues to block **585** and a machine-learned mosaic analysis and content browsing path are determined (as described in further detail below under the "Content Grid Detection" subheading). Then, at step **586**, the data generated by the mosaic analysis and content browsing algorithm is transmitted to the remote server.

If no mosaic is identified at step **584**, the process **580** of FIG. **5B3** continues at step **587**, and a determination is made whether any additional content information was identified in step **583**. If some useful content information (e.g., logos, objects, faces, etc.), the method continues to step **588**, and that data is transmitted to the remote server. If no useful content information was identified at step **587**, the method continues to step **589**, and a determination is made whether a predetermined period of time has passed since the last video frame capture and/or analysis. If the predetermined period of time has passed, the method returns to block **583**, and the machine-learned frame analysis is performed on the next frame. On the other hand, if the predetermined period of time has not passed, the method continues to step **590**, and a determination is made whether there has been change in the selected media source. If there has been no change in the selected media source, the method **580** returns to step **581** and waits for the next video frame. On the other hand, if there has been a change in the selected media source, the method **580** proceeds to step **591**, and the method then returns to step **530** of FIG. **5B**.

With reference now to FIG. **5B4**, an exemplary method associated with content identification rules for a video signal from a disc player or video game console is shown.

FIGS. **5A-5B4** illustrate a simplified exemplary operation of the system **100**. It will be appreciated that numerous additional steps have been excluded for the sake of simplicity. For example, in addition to continually monitoring the media input source, the system also continually monitors whether user registration has changed. Again, this is accomplished by monitoring input from the active registration buttons on the remote control **206**, as well as the passive registration techniques associated with the mobile devices associated with each panelist (e.g., smartphones, watches, bracelets, etc.). As explained in further detail below, when user registration information has changed, the information is overlaid on the content currently presented on the screen of the television **200**.

Although not shown as a particular step in the methodology of FIGS. **5A** and **5B**, it will be recognized that router capabilities of the gateway **110** allow it to also serve as a Wi-Fi sniffer that detects content consumed on other Wi-Fi-equipped devices (e.g., smartphones, tablets, desktop computers, etc.). In particular, simultaneously with monitoring the content presented on the television **200**, the gateway **110** is also configured to monitor Wi-Fi traffic at any of various devices connected to the gateway's Wi-Fi network. The content sniffed by the gateway **110** is tied to the panelist associated with the Wi-Fi-equipped device that presented the content. In this manner, all media content consumed at a

US 10,932,002 B2

21

household **202** is collected by the gateway **110** and associated with individual panelists within the household. Methods associated with detecting content presented at Wi-Fi enabled devices is described in further detail below with reference to FIGS. **18-20** under the heading “Internet Activity Measurement.”

By implementing the method of FIGS. **5A** and **5B**, the system **100** is equipped to improve upon conventional media content measurement devices. The system **100** implements unique hardware and software components and functionality in order to collect the appropriate data that is capable of providing an understanding of essentially all of the media content being consumed within a household and the specific panelists consuming such content.

FIG. **6** is an illustration showing a data series **600** collected from an exemplary household/family of panelists within a single day. The household includes the following panelists; (i) a 43 year old adult female, (ii) a 45 year old adult male, (iii) an 11 year old boy, and (iv) a 4 year old girl. As shown in the illustration, at some time between 9:00 am and 11:00 am, all of the panelists are present in the kitchen and the gateway identifies the program “CBS Sunday Morning” as presented on the kitchen television. The program airs for some period of time (e.g., from 9:15 to 11:00 am) and all panelists are identified with the show during this period of time. During this time a number of ads are also identified as been presented to the panelists. These ads include the Suave “Art Exhibit” ad, the Casper “Only Casper” ad, the Naked “Steps” ad, the Chipotle “Kitchen” ad, as well as a number of additional ads as shown in FIG. **6**. Also during this time, the gateway **110** identifies that content from the “Food Network” app was presented to the adult female on her device. The gateway also identifies that content from the “ESPN” app was presented to the adult male. The television is turned off around 11:00 am, at which time the 11 year old boy goes to the basement where he watches a “Spongebob Squarepants” episode in the basement using Roku via the Amazon Prime app. At some point after 11:30 am, the adult female and the 4 year old girl go to the bedroom and watch the PBS show “Daniel Tiger” via video on demand. The collection of data concerning consumed media content then continues throughout the day, until all devices are turned off by 10:45 pm. Advantageously, the data collected includes data describing the display device, where the device is located (i.e., if a non-mobile device such as a television), the platform and/or apps used to watch the media content (e.g., cable TV, Amazon Prime, etc.), the specific content watched (e.g., ads, programming, gaming, etc.), the specific the panelists watching the content, and the time the content was watched. The collected data for the household is periodically transmitted to the remote server **310** throughout the day.

FIG. **7** shows an exemplary set of data **700** collected from a number of different households on a particular day (e.g., Jan. 15, 2020). This data set **700** includes a number of fields **702** identifying the media consumed, a number of fields **704** identifying the panelist who consumed the data. Advantageously, the data may be processed in order to identify trends in the data. For example, the data may indicate that 25% of adults between the ages of 25 and 35 who subscribe to cable watched a particular network drama (e.g., “Game of Thrones”) on this day, and 80% of those viewers were exposed to a particular advertisement. The data set **700** may also be further processed to expose additional information and trends. For example, the data set **700** may indicate that 5% of the viewers exposed to a particular advertisement actually searched for the advertised content on their smartphone within ten minutes of viewing the advertisement.

22

Therefore, by collecting and analyzing data using the gateway **110**, advertisers, programmers, and others in the media industry are equipped to learn the level of exposure of programming and advertisements, and responses to such programming and advertisements. This also equips those in the media industry to make informed decisions with respect to future programming and advertisements.

Multi-Layered Approach to Content Recognition

As noted above the gateway **110** is configured to identify media content presented on a television **200**. The gateway does this by first identifying what source is providing the television input, and then applying different rules to determine the content based on the identified source. Because of the different rules associated with different input sources, the gateway **110** is considered to take a “multi-layered approach” to content identification.

To further illustrate the multi-layered approach, consider the exemplary arrangement of FIG. **8** wherein five input sources are connected to the I/O ports of the gateway **110**. The input sources include four HDMI inputs, including a cable/satellite box **212** connected to the HDMI-IN 1 port, a Blu-Ray/DVD player **214** connected to the HDMI-IN 2 port, a video game console connected to the HDMI-IN 3 port, and an OTT device connected to the HDMI-IN 4 port. These four HDMI inputs are fed to the video capture module **115** of the gateway **110**. The fifth input source is a digital TV antenna input **219**, which is fed to the video capture module **115**. The video capture module **115** includes various hardware and software components for processing the received signals, including an ATSC tuner **139** (which receives the antenna input **219**), an HDMI capture API **163** (which may be considered part of the content capture engine **162**), and various OTT apps **161**. Using the remote control **206**, the user selects one of the four input sources for presentation (i.e., display) on the television **200**. The video capture module **115** then outputs the video signal associated with the selected input source to the television **200** via the HDMI-OUT port of the gateway **110**.

The video signals delivered to the video capture module **115** via the four HDMI inputs are all industry standard video signals. The video signals delivered to the video capture module **115** via the antenna and the ATSC tuner **139** are ATSC or other standard broadcast signals. The video signals delivered to the video capture module **115** may be characterized as a series of still images called “frames” (or screen shots) that are delivered in rapid succession at a constant interval (i.e., frame rate). As the frames of video are received by the video capture module **115**, the frames from the selected input source are passed on through the HDMI-OUT port **133** and to the television **200** where they are then presented on the television screen. During this time, the video capture module **115** analyzes selected frames of the video signal presented on the television. The frames selected for analysis are determined by a specific set of content identification rules that are dependent on the selected input source. The content identification rules define the method for processing the associated video signal, including rules for identifying video frames for content identification analysis, and what procedures for content identification will be applied to the identified video frames. In some rule sets, each and every frame from a particular source may be monitored and, based on certain trigger events, selected frames may then be subjected to additional in-depth analysis. In other rule sets, only predetermined frames are selected for in-depth analysis following the occurrence of a trigger events. As used herein, the term “trigger event” refers to some occurrence indicating that a subsequent more in-depth con-

US 10,932,002 B2

23

tent analysis should be conducted on one or more video frames in an attempt to determine the media content presented on a screen device (e.g., the television 200). In at least some embodiments, the frames associated with a trigger event are not only analyzed, but also captured (i.e., stored in memory), and/or transmitted to the remote server 310 or cloud for additional processing.

A number of different trigger events are possible. In general, trigger events may be split into two categories: (i) video frame triggers (which may be referred to herein as “frame triggers”), and (ii) external triggers unrelated to the video frame (which may be referred to herein as “external triggers” or “non-frame triggers”). Frame triggers occur when a preliminary analysis is conducted on a video frame which indicates that some trigger event is happening. Frame triggers are often used in rule sets where the video frames associated with a video signal are regularly monitored (e.g., many, most or all of the frames are subjected to some preliminary frame analysis). A first type of frame trigger occurs when a programming grid or content banner is included in one frame of the stream of frames. For example, this trigger event may occur when the preliminary analysis of a video frame includes pixels indicative of content banner or programming grid (e.g., a box or grid structure overlaid on some portion of the screen that includes some programming information, including information for a single channel and/or information for multiple channels). A second type of frame trigger occurs when a scene change is detected in the video signal. For example, the preliminary analysis of the stream of video frames may include comparing consecutive frames in the video input in order to detect a temporary blank screen (e.g., five consecutive blank screens), or a threshold change in the pixel density from one frame to the next (e.g., as may be the case when the content changes from network programming to an advertisement). A third type of frame trigger occurs when a network logo appears or does not appear within a frame. Inclusion of a network logo is generally indicative of network program content. Similarly, a missing network logo may be indicative of advertising or other content that is not created by the network. While three frame triggers are mentioned herein, it will be recognized that numerous additional frame triggers are possible and contemplated for use in the field. As discussed in further detail herein, frame triggers are often provided by the analysis from one or more local machine learning engines configured to detect such frame triggers.

Unlike frame triggers, external triggers are not related to the video frame itself. A first type of external trigger occurs when an infrared signal (or other type of signal, such as a short-range RF signal) is sent to the gateway 110 from the remote control 206. The signal may be any number of different possible signals sent from the remote control 206, such as a channel change signal, volume change signal, input source change signal, menu signal, television guide signal, etc. A second type of external trigger occurs when a signal for the gateway 110 is received from a source other than the remote control, such as a user voice signal. Again, this signal may be any number of different signals associated with control of the television via the gateway. A third type of external trigger occurs is when the gateway 110 detects a change in panelist registration occurs (i.e., an individual considered to be consuming the displayed content) is detected. A fourth type of trigger event is the expiration of a threshold period of time since the last trigger event (e.g., 500 ms, one second, one minute, five minutes, etc.). This trigger event ensures that the input video signal is captured and analyzed at least periodically (e.g., every five minutes),

24

even if no intervening trigger events occur. While four types of external triggers are mentioned herein, it will be recognized that numerous additional external triggers are also possible. Additional information and descriptions of various triggers, including examples of external triggers and frame triggers, are provided below under the “Exemplary Trigger Events” subheading.

As noted previously, each set of content identification rules defines different signal monitoring procedures and trigger events. When a trigger event occurs, the video capture module 115 strategically analyzes one or more frames of the selected video signal as defined by the rule sets. The gateway 110 may define any number of different rule sets for content identification. For example, in the embodiment of FIG. 8, a first set of rules with a process flow similar to that of FIG. 5B1 is used if the selected video signal for display on the television is received from the antenna input/ATSC tuner 139. A second set of rules with a process flow similar to that of FIG. 5B2 is used if the selected video input signal is received from the satellite/cable box 212. A third set of rules with a process flow similar to that of FIG. 5B3 is used if the selected video signal is received from the OTT device 218. A fourth set of rules similar to that of FIG. 5B4 is used if the selected video signal is received from the Blu-ray player 214 or the gaming console 216. Following application of each rule set, a data package is generated that includes content identification data for the associated media content presented on the television. Because each rule set is different, it will be appreciated that each data package is also different.

One example of a content identification rule set procedure is now provided in the context of the selected video source being the satellite/cable box input 212 of FIG. 8, and the rule set being a first set of rules. This first set of rules is uniquely adapted to analyze frames of the video signal from the satellite/cable box 212 and assemble a data package for the analyzed frames. As noted previously, the rule set defines the method for processing the associated video signal, including rules for identifying video frames for content identification analysis, and the specific in-depth content identification algorithms that will be applied to the identified video frames. In this example, consider that the exemplary rule set defines a process flow that is somewhat similar to that of FIG. 5B2, but instead of analyzing each and every frame of the video signal for frame triggers, the rule set defines frame analysis windows that follow immediately after the occurrence of external triggers.

Each frame analysis window defines a short time period following a specific trigger event when one or more frames are captured and/or analyzed (e.g., one frame, two frames, ten frames, all frames, etc.). As an example, when the external trigger event is receipt of a remote control signal to change the channel, the rules define a frame analysis window that occurs between one and five seconds after the trigger event, and a frame is captured every 0.5 seconds during this time. The defined frame analysis window is based on the expected or possible occurrence of some content information being displayed on the television screen within the defined window. This content information may be displayed in any number of different ways, depending on the specific cable provider, such as different types of content grid (e.g., a banner overlaid along the bottom of the image, or a programming guide overlaid on some portion or most of the television screen), a simple display of text, or even audio played for the viewer (in the case of audio, the gateway is configured to store the audio as text). The first set of rules strategically defines this frame analysis window based on

US 10,932,002 B2

25

the particular input source (e.g., set-top box) connected to the gateway **110**, which may include a specific model of set-top box. The term “model” of set-top box may refer to a specific content provider (e.g., AT&T Uverse, DirectTV, etc.) and/or a specific model number and/or part number of the set-top box. For example, an AT&T Uverse box having model number 123456 may place a content box with a blue-colored background along the bottom of the screen between one and four seconds after the receipt of a channel change signal. In this case, the content identification rules for this set-top box defines an associated frame analysis window (e.g., one to four seconds following detection of the banner) that utilizes a machine-learned algorithm to detect the presence of the AT&T content banner in one of the video frames within frame analysis window.

In addition to defining a frame capture window, the first set of rules incorporates machine learning modules within the content identification engine **164**, and is configured to analyze an identified video frame (or multiple frames) and provide outputs that identify the content being watched on the screen. The outputs primarily include data identifying what is being watched on the television (e.g., program name, channel, time, etc.), but may also include additional information, such as identified logos, faces, characters, etc. Thus, the content identification engine **164** includes a number of different machine-learned algorithms, each of which implement a number of different tools/hidden layers. Examples of these tools include a text recognition tool (e.g., OCR), and other computer vision tools such as a logo recognition tool, a character recognition tool and/or a face recognition tool. Selected ones of these machine-learned algorithms may be executed in parallel to arrive at the defined content for a particular video frame.

The machine learning engine may also implement additional tools to arrive at the content identification data. For example the machine learning engine includes a text classification tool that identifies a category/field for all of the text extracted from a content grid from a particular provider (e.g., an identification that specific text is associated with a program name, program time, program description, channel, current time, etc.). The text classification tools are dependent at least in part on the particular provider (e.g., AT&T) and the expected position of certain information on a banner or other grid from such provider (e.g., the network is on the left side of the banner, the program name is in the middle of the banner, and the channel number is on the right side of the banner).

With reference now to FIG. 9, a screen shot **250** associated with an exemplary video frame captured by the gateway **110** is shown. FIG. 9 also shows content identification data **260** extracted from the video frame using the first set of rules described above (i.e., a rule set associated with the cable box **212**). As shown in the screen shot **250**, a programming banner **252** is overlaid on the program content **254** on a lower portion of the screen. The banner **252** includes a significant amount of text **256** that may be used to identify the program content. The machine learning engine reads the text **256**, splits the text into different blocks of text, and then categorizes each block of text (e.g., program name, program time, program description, channel, current time, other information). Again, the categories for each block of text may be derived based on the known source (e.g., AT&T Uverse cable box), and the known location of data within the AT&T Uverse content grid/banner.

In the example of FIG. 9, the rule set has extracted 184 characters of text from the frame, split the extracted text into

26

blocks, and identified data categories and associated data for each category, as shown in the following table:

TABLE 1

Data Category	Extracted Data
Program name	“Malcom in the Middle”
Program time	“3-3:30pm”
Description	“Lois’ Sister,” S5/Ep13, (2004), (TV-PG,L), Lois and her competitive sister must reconcile because Susan . . .”
Channel	885 FUSE
Current time	3:24 pm
Other information	3HDDOD, CatPG

While table 1 illustrates an example of extracted text from a frame of video, and categorization of such data in order to identify programming content, it will be recognized that additional data may also be extracted from a frame of video. Examples of such additional information include logos, faces, products (e.g., cars, shoes, etc.), image labels (e.g., houses, cars, trees, animals, etc.), or any other information that may be of assistance in content identification and that the machine learning engine is trained to recognize.

The content identification data is incorporated into a data package for each analyzed video frame. Each data package includes different information, depending on the input source of the content. For example, different types of additional information associated with different input sources may include a viewing source, viewing type, viewing platform/provider, application name, and program type, and system information from the time of capture. The input source may include, for example, paid TV, OTA TV, recorded content, streaming content, video game, mobile source, etc. The viewing type may include, for example Live TV, Playback, video on demand (VOD). The viewing platform/provider may include, for example, AT&T set top box, Comcast set top box, Xbox console, PlayStation console, AppleTV, Amazon Firestick, etc. The application name may include, for example, Netflix, Amazon Prime, Hulu, etc. The program type may include, for example, TV program, TV ad, movie, video game, etc. System information from the time of capture includes, for example, a timestamp, TV on/off status, etc. Additional information on various data packages is provide below under the “Data Packages” subheading.

Exemplary Trigger Events and Content Identification Algorithms

A more detailed explanation of various trigger events and associated content identification programs are now described with reference now to the exemplary triggers illustrated in FIGS. 10A-10L. As discussed previously, numerous sets of content identification rules are stored in the gateway, and each set of content identification rules includes one or more defined triggers. The gateway **110** applies one set of content identification rules to the video signal output to the television. The applied set of content identification rules is based on the selected input (i.e., the input to the television **200**, which is the output of the gateway **110**).

Infrared External Trigger Event

FIG. 10A illustrates a first exemplary external trigger event in the form of an infrared signal from a remote control. The trigger event is included with a particular set of content identification rules associated with a particular cable box (e.g., a Comcast cable box). In the example of FIG. 10A, the infrared signal is received from the gateway’s remote con-

trol **206**, but it will be appreciated that the trigger event could be defined by an infrared signal from any remote control.

As shown in FIG. **10A**, when the user presses a button on the remote control **206**, an infrared signal is transmitted. The gateway **110** receives the infrared signal and performs the requested action (e.g., a channel change) on the selected source/device. When the gateway **110** receives the infrared signal, the content capture engine **162** performs a capture and/or analysis of one or more frames of the video signal that are presented on the television follow the trigger event. Again, the content identification rules define the number of video frames to analyze, what analysis should occur, and the timing of the analysis.

In the example of FIG. **10A**, consider a situation where the content identification rules indicate that one frame of the video signal should be captured and analyzed every 250 ms within a frame analysis window of three seconds. Based on these rules, the gateway **110** will capture 12 frames over the three seconds (i.e., 4 frames/second \times 3 seconds=12 frames) that follow the trigger event. The content identification rules also indicate that each of these frames should be analyzed with the grid detection algorithm. When a grid is detected (e.g., in the form of banner **252** of FIG. **10A**), the rules then indicate that the text from the grid should be subjected to a content extraction algorithm that is unique to banners associated with the particular content provider and device (e.g., the specific type of Comcast cable box connected to the gateway). The content extraction algorithm not only identifies text within the banner, but also categorizes the identified text. For example, the content extraction algorithm may determine blocks of text as indicating particular data based on any number of parameters such as the location of the text within the banner, the font of the text, the format of the text (e.g., a time format), proximity of the text to various indicia (e.g., a program timeline, icons, etc.), and any of various other parameters that the machine learning model determines to be significant. In at least some embodiments, the categorization portion of the content extraction algorithm is a machine-learned algorithm. In other embodiments, the categorization portion is a human-programmed algorithm.

In addition to identifying text, and classifying the text, the content identification rules further indicate when the content identification process should be terminated (e.g., prior to the full three second period). For example, in the embodiment of FIG. **10A**, the content identification period may terminate when either (1) two consecutive video frames are analyzed and identify the same content, or (2) a specific additional trigger event occurs (e.g., a new channel change signal from the remote control). When content is identified, the data may be saved in the internal memory of the gateway **110** and/or transmitted to the remote server **310**. It is also possible that the rules may instruct the gateway to discard (and/or do not transmit) the identified content the content was not presented on the television for a threshold period of time (e.g., a subsequent channel change signal was received within ten seconds).

When the content identification rules are executed in the example of FIG. **10A**, no grid may be detected for the first few frames that are captured following the trigger event (e.g., it may take 1 second for the content banner **252** to appear on the television **200**). Accordingly, the content identification process does not occur for the frames captured and analyzed during second one of the frame analysis window. However, if the banner **252** then appears during second two of the frame analysis window, the gateway

detects the occurrence of the banner **252**, and the content identification rules applies a content extraction algorithm to the banner. The content extraction algorithm not only recognizes text, but also categorizes the text in order to determine specific data taken from the banner. For example, in FIG. **10A**, it may be determined that the text "Criminal Minds" is indicative of a program name and the text "2007" is indicative of a year when the program first aired. In this case, when data from two (or more) consecutive frames are identical, the rules instruct the gateway to terminate the analysis process, and the data associated with the identified content is saved and/or transmitted to the remote server. Because the data associated with each set of content identification rules is slightly different, it will be recognized that the data packages from different rule sets will also be different. Various examples of such data packages are described in further detail hereinafter with respect to FIG. **21A** under the "Data Packages" subheading.

While FIG. **10A** provides one example of an external trigger event and content identification rules associated therewith, it will be recognized that numerous variations of such rules and trigger events are possible. For example, in at least one embodiment, when an infrared signal is detected, the gateway **110** performs a media capture of the current incoming media, and all captured frames are stored in the gateway's memory without any analysis. Thereafter, the captured frames are transmitted to the remote server/cloud for analysis. In at least some embodiments the rules call for capture and/or analysis of all identified video frames within a frame analysis window, even if there is no useful information in them. For example, in the case when frame captures occurs because a user changes the volume level with his remote control, the captured frames may or may not contain useful information. On the other hand, when a user changes the channel, the captured frames have a high probability of containing useful information such as the program name, channel number, network name or other useful information.

Time-Based Frame Captures

FIG. **10B** illustrates a second exemplary external trigger event defined simply by the passage of time. The trigger event is included with a particular set of content identification rules associated with a particular game console (e.g., a Sony PlayStation). In the example of FIG. **10B**, the content identification rules indicate that a screen should be captured and/or analyzed every ten seconds. Accordingly, video frames are captured for content **410** displayed on the television **200** at time 10:05:20, content **420** displayed on the television **200** at time 10:05:30, and content **430** displayed on the television at time 10:05:40. Subsequent video frames are also captured every ten seconds for the entire time that content from the game console is displayed on the television (e.g., frame captures every ten seconds for an hour or other time that the game console is in use).

The content identification rules identify specific in-depth content analysis procedures to be performed on each screen captured. As noted previously, in-depth content analysis is often based on machine learning models. For the video game console of FIG. **10B**, the content identification rules may call for machine-learned algorithms, such as text recognition, logo/trademark recognition, character recognition, object identification, etc. These machine-learned algorithms are run sequentially or in parallel for each captured frame, as defined by the content identification rules. In some instances, these machine-learned algorithms are complex and require significant processing power. Accordingly, for these algorithms, the captured video frames are transmitted

to the cloud/remote server **310** for further processing. In any event, the content data returned from the local and/or remote content identification engines provides valuable information concerning the particular content being played on the television. For example, in the example of FIG. **10B**, the content identification engine may determine that the user is playing the “Call of Duty: Advanced Warfare” game. As another example, in some instances the content identification engine may simply identify video game content in a generic manner, such as: “video game/war game.” Again, the data collected based on the content identification rules is assembled into a unique data package and saved to the gateway **110** and or transmitted to the cloud/remote server **310**.

While FIG. **10B** illustrates one example of a time-based external trigger event applied on one set of content identification rules, it will be recognized any number of additional time-based trigger events are possible. For example, in at least one embodiment, a time-based trigger event could require a frame capture every one minute with programming from a cable box, if no other trigger event occurred within the past minute. In yet another embodiment, a time-based trigger event could occur every five seconds with OTA content in an attempt to capture all advertising content displayed while a panelist is watching a broadcast channel.

Content Grid Detection

FIGS. **10C** and **10D** illustrates a first exemplary frame trigger event based on content grid detection, wherein the content grid is provided in the form of a content banner **252**. The trigger event is included with a particular set of content identification rules associated with a particular cable box or satellite box (e.g., Comcast cable box, Direct TV satellite box, etc.; the term “set-top box” as used herein refers to either a cable box or a satellite box). In the example of FIGS. **10C** and **10D**, the content identification rules analyze each and every frame of video from a video signal and determine whether a content banner **252** is present on the television **200** (as noted on the left side of FIG. **10C**) or is not present on the television **200** (as noted on the right side of FIG. **10C**). When the content banner **252** is present, the content identification engine **164** performs further processing on the identified frames in order to identify the content provided within the banner (in a similar manner to that described above with reference to FIG. **10A**).

FIG. **10D** shows an exemplary log file **450** of the gateway **110** based on the analysis of a single video frame (and particularly a video frame with a content banner, such as that shown in FIG. **10A**). The log file illustrates the steps taken by the gateway **110** when implementing a set of content identification rules, and particularly those associated with extracting programming data from a content banner. At line **451**, the log file **450** shows that the gateway was reviewing video frames and waiting for a content grid to appear in one of the frames. At line **452**, the log file **450** shows that a content grid in the form of a banner was detected in one of the frames. At line **453**, the log indicates that further processing confirmed the presence of the content banner **252** within the frame. Accordingly, a trigger event is shown in lines **452** and **453** by the detection of a content banner. This trigger event resulted in further processing on the frame, as noted in lines **454-458** of the log **450**, in order to identify the content associated with the frame. In particular, at line **454**, the log **450** indicates that the process of extracting text from the grid was performed (e.g., via a text recognition tool such as OCR). At line **455**, the text categorization process identified the name of the program as “Criminal Minds (2007)”. At line **456**, the text categorization process identified the

network as “ion HD”. At line **457**, the text categorization process identified the channel number as “**531**”. Then, at line **458**, the log **450** indicates that a data package with this information is assembled and transmitted from the gateway **110** to the remote server **310**. Finally, at line **459**, the log **450** shows that the gateway returned to reviewing video frames for another trigger event in the form of detection of another content banner. This process of extracting text from a grid and categorizing or otherwise identifying such text is one example of a content extraction algorithm defined by a set of content identification rules.

While FIGS. **10C** and **10D** illustrate a frame trigger and related processing based on the detection of a content banner, it will be recognized that other forms of content grids and subsequent processing are also possible. For example, as shown in FIG. **10E**, content grids may also take the form of content mosaics **352**. Content mosaics **352** include numerous blocks **354** of content (or content sources) that are simultaneously presented on a screen. The blocks **354** may be rectangular or any other shape, but they are selectable by a viewer in order to lead the viewer to desired content for presentation on the screen. Using a remote control, the user is able to move to any block on the mosaic by moving a selector. The current position of the selector (i.e., the block that the selector is current associated with) is highlighted in some way for the viewer. For example, the current block associated with the selector may be highlighted by an enhanced border, enhanced shading, an enlarged block relative to neighboring blocks, or any of various other highlighting techniques. As the user moves the selector from block to block, each selected block is highlighted. When the user wishes to view the content identified in the highlighted block, the user selects the block by taking an appropriate action, such as selecting an enter button on the remote control (e.g., the “OK” button).

The gateway **110** is configured to identify content presented on the television screen based on a user’s manipulation of a content mosaic **352**. FIG. **10F** shows an exemplary log file **460** of the gateway **110** based on the analysis of a single video frame, and particularly a video frame with a content mosaic **352**, such as that shown in FIG. **10E**. The log file **460** illustrates the steps taken by the gateway **110** when implementing a set of content identification rules, and particularly those associated with extracting programming data from a content mosaic. At line **461**, the log file **460** shows that the gateway was reviewing video frames and waiting for a content mosaic to appear in one of the frames. At line **462**, the log file **460** shows that a content grid (in the form of a mosaic) was detected in one of the frames. At line **463**, the log indicates that further processing confirmed the presence of the content mosaic **352** within the frame. Accordingly, a trigger event is shown in lines **462** and **463** by the detection of a content mosaic. This trigger event resulted in further processing on the frame, as noted in lines **464-468** of the log **460**. In particular this further processing identified content selected from the mosaic by the viewer. At line **464**, the log **460** indicates that the process of extracting text from the mosaic was performed. At line **465**, the log indicates that the user selected one of the blocks of the mosaic. At line **466**, the log indicates that the text from the selected option was extracted. At line **467**, the text identification tool identified the selected option (e.g., “TV”, “Game of Thrones”, “Friends: The One Where Everybody Finds Out”). Then, at line **468**, the log **460** indicates that a data package with this information was assembled and transmitted from the gateway **110** to the remote server **310**. Finally, at line **469**, the log **460** shows that the gateway

31

returns to reviewing video frames for another trigger event in the form of detection of another content banner.

As illustrated in the foregoing examples, the gateway 110 maintains a log of various screens presented to a user and various selections made by the user. This log in combination with the various content identification rules allows the gateway 110 to actually track a user's path/journey as they make their way through various content options, including input sources, menus, mosaics, and any of various other options presented on the television screen. The information on such paths is extremely valuable to content providers because it can help content providers understand how to most effectively deliver content to consumers.

FIG. 10G illustrates an example of such a log 470 showing a user's path to selecting and watching a movie. As noted at the top of FIG. 10G, the user has selected the HDMI2 option for input to the television 200 from the gateway 110. In this case, the HDMI2 option is a Roku device. As noted in line 471, the user's journey began at the main menu of the Roku device. As noted in line 472, the user then selected the Disney+ App from the main menu of the Roku device. Line 473 shows that the user next browsed the main menu of the Disney+ App. As noted in line 474, the user then selected the Marvel category from the Disney+ menu. At line 475, the user browsed the "Marvel" menu. At line 476, the user selected the "Iron Man" category from the "Marvel" menu. As noted in line 477, the user chose the "Iron Man 3" movie. Then at line 478, the movie started. This simple example shows how a user's path through content may be tracked. The gateway can be configured to track this path at any level of detail, including for example, user movement through various input sources, menus, and blocks of a content mosaic. With this information in hand, the most advantageous positions within menus, mosaics, and other content display platforms may be determined.

Detection of Network Logos

FIGS. 10H and 10I illustrate another exemplary frame trigger event based on detection of network logos 258. Once again, the trigger event is included with a particular set of content identification rules associated with a particular input source (e.g., Comcast cable box). In the example of FIGS. 10H and 10I, the content identification rules analyze multiple frames per second (e.g., 30 frames per second) from a video signal and determine whether a network logo 258 is present on the television 200, or is not present on the television 200. While detection of network logos is a machine-learned process, this processing can typically be performed on the gateway because of the limited number of network logos currently in use (e.g., less than one thousand). The network logos that may be detected by the gateway 110 include the network logos that appear on any of various programming content, including that delivered by cable/satellite television providers (e.g., AT&T, Comcast, DirectTV etc.), OTT providers (e.g., Netflix, Amazon Prime, Hulu), video game console brands and game titles (e.g., Xbox, PlayStation), and various other providers.

Advantageously, the content identification rules may implement the network logo detection algorithm in various situations. Primarily, the presence of the network logo indicates the currently tuned network (or OTT provider, video game console, etc.). Identification of a network logo can improve the efficiency and accuracy of content detection by providing confirmation that other identified content is correct (e.g., that the content extracted from a grid is correct). Furthermore, the absence of a network logo in the transmission may also be valuable. For example, the absence of a network logo could mean that a television advertise-

32

ment/commercial is currently being broadcasted (logos are not typically present in commercials). Accordingly, the network logo detection algorithm may serve as a trigger event to run other content detection algorithms that are more associated with advertising (e.g., see the discussion below under "Brand Detection"). Alternatively, the absence of a network logo may indicate other activity, such as the user navigating a content grid, the user browsing other services provided by the cable operator, or that the cable set top box is on standby mode or displaying a screen saver. Thus, it will be recognized that detecting logo absences is also an important process in order to improve the efficiency and accuracy of the content recognition.

FIG. 10I shows an exemplary log file 480 of the gateway 110 based on the analysis of a single video frame (and particularly a video frame with a network logo 258, such as that shown in FIG. 10H). The log file illustrates the steps taken by the gateway 110 when implementing a set of content identification rules that includes determining the existence of a network logo 258. At line 481, the log file 480 shows that the gateway was reviewing video frames and detecting the existence of network logos in each frame. At line 482, the log file 480 shows that a network logo was detected in one of the frames. At line 483, the log indicates that further processing identified the logo as the "FOX SPORTS" logo. At line 484, the identified network logo was included in a data package and transmitted from the gateway 110 to the remote server 310. At line 485, the gateway continued reviewing frames for presence of a network logo. At line 486, a frame was identified that did not include a network logo. At line 487, the absence of a network logo was noted and transmitted to the remote server. As noted above, the absence of a network logo in a video frame could itself serve as a trigger event, causing the content identification rules to perform subsequent processing, such as analysis of advertisements.

Real-Time Scene Change Detection

FIG. 10J illustrates another exemplary frame trigger event based on detection of scene changes. Once again, the trigger event is included with a particular set of content identification rules associated with a particular input source (e.g., Comcast cable box). In the example of FIG. 10J, the content identification rules analyze all frames of the video signal and determine whether a scene change has occurred. A scene change may indicate a number of different events, including a transition to advertising, a channel change, or simply a new scene within a program. Advantageously, the machine-learned scene change algorithm is capable of identifying a scene change, and the content identification rules are configured to use the scene change as a trigger event for further processing (e.g., analyze for advertisement, new channel info, etc.).

The scene change algorithm may be configured to detect a scene change in a number of different ways. For example, the algorithm may monitor a series of consecutive frames and look for some threshold number of consecutive blank frames (e.g., three frames) within the series. In FIG. 10J, a series of consecutive frames 280 is shown with five consecutive blank screens 281 included within the series 280. The content identification rules detect this series of blank screens 281 as a trigger event to implement an advertisement identification algorithm which includes several in-depth machine processes, such as face recognition and/or brand logo recognition.

When a scene change is detected, the gateway immediately stores a timestamp of the event, and additional processing begins as defined by the selected content identifi-

cation rules. When a brand is detected within the video frames following a scene change, the machine-learning model appends the scenes associated to that particular television ad, in order to obtain the final duration of the ad, start and end time. Other exemplary elements that may be detected following a scene change include landmark detection, persons detections, object detections, etc. A text version of the detected items may be stored in the database and incorporated into the data package output from the content identification rules. In at least some embodiment, the audio associated with a given advertisement may be captured from the HDMI signal and converted to text for storage in the local or remote database. This allows for further identification of advertisements in the event there is some question about what particular ad was presented on the television (e.g., the Coca-Cola ad with the polar bear, or the Coca-Cola ad with the puffin) when an advertisement occurs.

Another advantageous feature of the scene change detection algorithm is the ability to assign a length of time to the advertisement. For example, by reviewing a series of consecutive frames, the gateway **110** could register that a television commercial for Coca Cola ran from 11:23:30 to 11:24:00.

The scene change detection algorithm may also be used with other algorithms to specify a particular event (e.g., type of content change) occurred following a scene change. For example, if the frames immediately following a scene change do not include a network logo, the content identification engine may determine that a commercial is being played, and any of various advertisement identification algorithms may be run. On the other hand, if the frames immediately following a scene change still include a network logo, it is unlikely that the scene change was to a commercial, and the advertisement identification algorithms need not be run.

Brand Detection

The machine learning models further include algorithms configured to detect brand/trademark presences within content (e.g., Coca-Cola, Pepsi, Toyota, Ford, etc.). The detection of brands may be as simple as extracting text from a frame and identifying a particular string of text as a known brand. However, more complex machine-learned algorithms may also be used to identify brand logos. For example, in the example of FIG. **10K**, the machine-learned brand recognition algorithm may be configured to not only extract the text "Coca-Cola" **283** from the video frame, but may also be configured to recognize the ribbon **285** as a brand logo for Coca-Cola. With certain rule sets, brand detection may be used as a frame trigger (e.g., to indicate that a commercial is airing). However, in most rule sets, brand detection is used for more in-depth content analysis following another trigger event. For example, if a detected scene change (or alternatively, the absence of a network logo) indicates that a commercial is occurring, various in-depth brand detection algorithms may be applied to the video frames following the scene change as part of an advertisement identification algorithm. In addition to identifying brands, the advertisement identification algorithms may also include other machine-learned algorithms that detect other components of a commercial. For example, the advertisement identification algorithms may indicate that a particular Coca-Cola commercial included people, a beach, and a dog. This identified content is of great value in determining specific advertising content that was viewed by specific panelists at a specific time.

In view of the above, it will be recognized that many of the trigger events associated with FIGS. **10C-10K** are frame

triggers associated with internal machine learning models and algorithms. While some of these algorithms that provide the frame trigger (e.g., grid detection) implement relatively simple machine-learned algorithms, others are much more complex (e.g., face recognition) and require significant processing power. Thus, while each of the machine-learned algorithms described above may be used as trigger events, the more complex machine-learned algorithms are more appropriate for in-depth content analysis than as frame triggers. Accordingly, it will be appreciated that different rule sets within the gateway will implement the machine learning models and algorithms differently. For example, one rule set may implement a logo recognition algorithm as a trigger event, while another rule set may only implement the logo recognition algorithm following a trigger event.

Changes in Signal Parameters of an OTA Content

The content identification rules for over-the-air (OTA) content is generally distinct from those used with any other input sources. When a user is watching OTA television, the digital tuner is constantly reading the incoming digital signal coming from the antenna. OTA transmissions inherently include specific content information such as network name, channel number and program name in their signal. Whenever the gateway **110** detects a change in these parameters (i.e., the content information in the OTA transmission), the gateway determines that the user has changed a channel. At that time, the gateway reads the parameters and registers them. In this way, the OTA transmission itself can serve as an external trigger (i.e., a non-frame trigger) causing the gateway to record new information each time the parameters of the OTA transmission changes. However, it will be recognized that the content identification rules for OTA content may also include any of various additional trigger and/or machine-learned algorithms discussed above. For example, the frames of an OTA transmission may be periodically reviewed (e.g., every five seconds) to determine if a network or brand logo is present. Alternatively, the frames of OTA content may be continually reviewed to determine whether a scene change occurred.

Machine Learning Modules and Training

As noted above, the gateway **110** makes use of numerous machine-learned algorithms (which may also be referred to herein as "machine-learning modules") within the various content identification rule sets. Each of these machine learning modules is trained remote from the gateway to perform a particular task. The trained modules are then transmitted from the cloud to the gateway **110** and stored as machine learning modules within the content identification engine **164**. The content identification engine **164** implements these modules and provides various forms of content identification data as an output.

Each of the machine learning modules is trained to perform a particular task. For example, the content grid identification module may be trained to detect the occurrence of a content banner (or content grid) within a video frame. As another example, the network logo identification module may be trained to detect the occurrence of network logos in a video frame. In any event, the training process includes first creating a training set comprising a plurality of video frames. The training set includes a first plurality of video frames that include the occurrence of defined content (e.g., a content grid, network logos, text within a content grid, etc.) and a second plurality of video frames that do not include the occurrence of the defined content (e.g., no content grid, no network logos, etc.). The defined content may be, for example, any of the above defined frame triggers

US 10,932,002 B2

35

(e.g., content grids, network logos, brands, scene changes, etc.). The training set is then used to train a machine-learned algorithm configured to detect the occurrence of the defined content within a video frame.

The machine-learning modules may be trained as neural networks with various layers (e.g., input layer, hidden layers, output layer) and nodes within each layer, as well as various weights applied to the nodes within each layer based on the training. It will be recognized that any number of different parameters and weights may be assigned to the various nodes in order to arrive at the probabilistic output. Moreover, the weights and nodes may be adapted over time as iterative training occurs. Because the content data output from the machine learning engine is actually a high probability prediction of content, the reliability of the identified content improves over time with additional learning. This additional learning typically occurs remotely and is transmitted periodically to the gateway 110 in order to update the various machine learning modules.

The output of the machine-learned algorithm indicates the occurrence or non-occurrence of the defined content within the video frame. The output of a machine-learning module within a given rule set may serve to generate data related to the media content itself (e.g., brands, logos, etc.), and/or may serve as a trigger event for implementation of another machine-learning module (e.g., the existence of a content grid).

Various machine-learning modules within the gateway 110 may be utilized to perform a first round of content recognition on the captured media. If the gateway's content identification rules are able to resolve the content with enough accuracy, the generated content data is packaged and transmitted to a remote location where it is stored in a cloud-based database. On the other hand, when the gateway's content identification rules do not resolve the content with sufficient accuracy (e.g., there are missing or unknown pieces in the data package), the data package and any associated video frames may be transmitted to a remote location for further processing. This remote processing typically includes much more powerful machine learning modules that are not efficiently run at the gateway. For example, face recognition, character recognition, advanced game recognition modules, etc. may all be more efficiently performed with remote cloud-based software. The output of these modules may then be used to complete or supplement any data packages generated using the gateway's content identification rules.

In view of all of the foregoing, it will be appreciated that the gateway includes various machine learning models that perform real-time detection of elements included in the frames of an incoming video signal. Models are initially trained using cloud computing infrastructure, which provides high computing power. Once the models are trained, they are deployed into the gateway 110 for it to perform recognition on the incoming video frames. Additional remote machine learning modules may be applied to captured video frames that are transmitted to the remote server 310 or other remote locations.

Active and Passive Panelist Registration

As noted above, the gateway 110 is configured to register panelists who are determined to be watching the television 200 connected to the gateway 110 at the time of registration. The gateway 110 is also configured to de-register panelists after some period of time when it is determined that the panelist is no longer watching the television 200. When a panelist is registered, that panelist is associated with the media content presented on the television during the period

36

of registration. In particular, when content is analyzed in order to identify the content (e.g., using the methodologies discussed in the previous section), all the panelists registered with the gateway 110 at the time the content is presented on the television 200 are also associated with the data that identifies the content. Therefore, the data generated by the gateway 110 not only identifies content presented on the television, but also identifies all panelists who watched/consumed the content.

Registration of panelists at the gateway occurs by either active or passive registration. Active registration requires conscious actions from at least one panelist in order to register the panelists who are currently watching the television 200. In at least one embodiment, active registration of panelists occurs by the user pressing one or more buttons representing the panelists to be registered. These buttons may be presented in several ways, including physical buttons on the remote control 206 for each panelist, physical buttons on the gateway housing 112 for each panelist, and virtual buttons overlaid on the television screen 200 when a registration menu is activated.

In a first embodiment, the remote control includes a dedicated button for registration of each panelist. For example, as shown in FIG. 11, the remote control 206 for the gateway 110 includes a series of differently colored buttons 208 positioned along the bottom of the remote control 206. Each of these buttons 208 is associated with one of the panelists 204 when the gateway 110 is initially set up for the household 202. For example, if the buttons 208 include a red, green, yellow and blue button, the red button may be associated with a first adult female, the green button may be associated with a first adult male, the yellow button may be associated with a first child, and the blue button may be associated with a second child. Thereafter, when the television is on and one of these buttons is pressed, the panelist associated with that button is registered with the device at that time. An indication of panelist registration may then be overlaid on the television screen (e.g., an avatar for the panelist may be presented on the screen along with a welcome message, as described in further detail below), or may appear on the gateway display 152. When the panelist de-registers, an indication of de-registration may also be overlaid on the screen (e.g., an avatar for the panelist may be removed from the screen and a good-bye message presented).

The buttons 208 may be configured in any of various ways to register and de-register panelists. For example, in one embodiment, each of the buttons 208 is a toggle switch such that pressing a button the first time registers the associated panelist, and pressing the button a second time de-registers the associated panelist. In another embodiment, the number of times the button is pressed within a short period of time registers or de-registers the associated panelist (e.g., one press within two seconds registers the associated panelist, and two or more presses within two seconds de-registers the associated panelist). Panelists effectively use the buttons 208 by registering when they enter a room and begin watching media presented on the television 200, and then de-register when they leave the room or otherwise stop watching TV. Again, during a panelist's viewing session, all detected/captured content information will have the panelist's identification and current timestamp attached to it, in order to associate the panelist, to the viewed content.

While the foregoing paragraphs describe one exemplary embodiment of active registration, it will be recognized that other forms of active registration are contemplated. For example, active registration may occur using physical but-

US 10,932,002 B2

37

tons provided on the gateway or virtual buttons provided on the television screen when a user enters a registration routine (e.g., a routine that may be called up using a physical button on the remote or virtual button in another menu). As yet another example, active registration may occur using voice commands provided to the gateway **110** (e.g., “Hey Google, Adam is watching television”).

In addition to active registration, the gateway **110** also provides for passive registration of panelists. Passive registration occurs automatically without any conscious effort by the panelist to register. The gateway **110** is generally configured to perform passive registration by detecting use proximity to the gateway **110** (and thus the proximity to the television **200**). In at least one embodiment, passive registration occurs by detecting the signal strength at the gateway **110** from mobile devices that are in communication therewith and associated to particular panelists. The mobile devices may be provided (i) by mobile computing devices such as a smartphones, tablets, watches, or other mobile devices configured with Wi-Fi communications capabilities, and/or (ii) by wearable devices with short range wireless communications capabilities such as dedicated watches or bracelets with Bluetooth communications capabilities, or any of various other devices with Bluetooth communications capabilities, such as earbuds.

Mobile computing devices within a household are registered with the gateway **110** (i.e., identification data is shared and the device is configured for automatic connection to the gateway) at the time of gateway setup (or any time thereafter). Registrations of smartphones and smartwatches with the gateway are particularly advantageous because these mobile computing devices are typically carried by the panelist at all times. In any event, the unique identifier/MAC address of a mobile computing device that is associated with a panelist is stored in the panelist data **172** of the gateway **110**. Each panelist's mobile computing device is configured to automatically connect to the same Wi-Fi network as the gateway **110**. As discussed in further detail below, the gateway's Wi-Fi chipset **125** allows the gateway **110** to either (i) serve as a router and establish a new Wi-Fi network, or (ii) operate in a sniffer mode in order to detect network traffic within the existing Wi-Fi network. In either case, the gateway constantly scans for MAC Addresses from devices communicating over the Wi-Fi network. When the gateway recognizes the MAC Address of a mobile device associated with a panelist, the signal strength (e.g., RSSI) of that mobile device is recognized to determine the proximity of the panelist to the gateway. When the signal strength from the mobile device is greater than a threshold strength, the panelist is determined to be in proximity to the gateway **110** and the television **200**, and the panelist is registered at the gateway. All identified presented on the television **200** is then associated with the panelist during registration. When the signal strength from the mobile device is less than the threshold, the panelist is determined to be outside proximity to the gateway **110** and television **200**, and the panelist is not registered with (or is de-registered from) the gateway **110**.

With reference now to FIG. 12, a table **1200** of WPA handshakes collected at the gateway **110** is shown. The table includes a list of MAC addresses associated with WPA handshakes between various mobile devices and the wireless access point (e.g., provided at the gateway **110** or household router). The BSSID column **1210** shows the unique identifier/MAC address for a number of devices communicating over the wireless network. Other information is also shown in the table **1200**, including a signal strength column **1220** which provides a numerical value representative of the

38

signal strength of the mobile device at the gateway **110**. When the signal strength is greater than a threshold value (e.g., 25), the panelist associated with the mobile device is determined to be in proximity to the television **200** (e.g., 25 feet), and the panelist is registered at the gateway **110**. When the signal strength is less than the threshold, the associated panelist is determined to not be in proximity to the television **200**, and the panelist is not registered (or is un-registered) at the gateway **110**. Advantageously, the threshold signal strength may be different for each gateway **110**, depending on the size of the room where the gateway is installed. For example, in a first home, the gateway may be installed in a room that is 20 ft×20 ft, and in a second home the gateway may be installed in a room that is 30 ft×30 ft. Because of this, a panelist determined to be 25 feet away from the gateway in the first home is unlikely to be in proximity to the television, while a panelist determined to be 25 feet away from the gateway in the second home is likely to be in proximity to the television. Accordingly, when the gateways are initially configured in these two homes, the technician or user may make the threshold in the first home to be less than the threshold in the second room.

FIG. 13 is a flowchart summarizing the above-described method **1300** of registering a panelist based on the signal strength of a MAC address. The method begins at step **1310** when the gateway **110** is turned on and the gateway joins a household Wi-Fi network (or serves as a router therein). At step **1320**, the method continues by conducting MAC address polling via the Wi-Fi beacon. At step **1330**, a list of detected MAC addresses is analyzed (e.g., similar to that of FIG. 12). At step **1340**, a determination is made whether a MAC address associated with one of the panelists is in the lists. If a MAC address associated with a panelist is in the list, the method continues to step **1350**, and the power signal level associated with the MAC address is analyzed by comparing it to a threshold. As step **1360**, if the power level is greater than the threshold, the associated panelist is considered to be in proximity to the television. On the other hand, if the power level is less than the threshold, the associated panelist is considered to be outside proximity to the television. At step **1390**, when the panelist is determined to be in proximity to the television, the panelist is registered and an avatar for the panelist is shown on the television. However, if the panelist is not in proximity to the television, the method moves to step **1370**, and a determination is made whether or not the panelist is currently registered with the gateway. If the panelist is not currently registered at step **1370**, the method returns to step **1320** and continues to poll for MAC addresses. However, if the panelist is currently registered at step **1370**, the method continues to step **1380**. At step **1380**, the panelist is de-registered at the gateway, and no avatar for the user is shown on the television.

As noted previously, in addition to passive registration based on the proximity of Wi-Fi-enabled mobile computing devices to the gateway, passive registration may also occur based on proximity of wearable electronic devices with short range wireless communications capabilities (e.g., Bluetooth) to the gateway. FIG. 14 shows an example of dedicated wearable electronic devices in the form of bracelets **1410** configured to be worn on the wrist and pendants **1420** configured to be worn around the neck or carried in the pocket of the panelist. Dedicated wearable devices are typically reserved for use by children or the elderly who do not own a smartphone or other mobile computing device capable of communicating with the household wireless network. However, dedicated wearable devices **1410**, **1420**, may also be used by adults who do not own a smartphone or

do not regularly carry their smartphone around the house. Other examples of wearable electronic devices include headphones, earbuds, or other wearable speaker devices (e.g., iPods), with short range wireless communication capabilities. These devices are increasingly being carried by individuals at all times, and may be used to passively register panelists with the gateway **110**.

Registration of panelists carrying wearable electronic devices is similar to registration of panelists with Wi-Fi-enabled mobile computing devices. However, instead of monitoring signal strength, the Bluetooth chipset **123** of the gateway **110** simply polls for Bluetooth signals from various wearable electronic devices. In particular, the gateway **110** continually sends a polling signal to ask whether any wearable devices are receiving Bluetooth signals from the gateway. The strength of the polling signals is such that only wearable electronic devices within a certain range (e.g., 25 feet) will receive the signal. The strength of the polling signal may be adjusted within the gateway **110** (e.g., by a technician via the gateway's configuration interface) in order to provide the proper signal range. When a response to the polling signal is received from one of the wearable electronic devices, the gateway recognizes the wearable device, and the associated panelist is registered at the gateway **110**.

FIG. **15** is a flowchart summarizing the above-described method **1500** of registering a panelist based on Bluetooth communications with a wearable electronic device. The method begins at step **1510** when the gateway **110** is turned on and the gateway Bluetooth chipset **123** is powered up. At step **1520**, the method continues with the Bluetooth transceiver sending Bluetooth polling signals, and listens for responses from any Bluetooth devices that are within range of the polling signal. At step **1530**, a determination is made whether any confirmation signals have been received from wearable devices in proximity to the gateway **110**. If a confirmation signal is received from a wearable device, the method continues to step **1540** where the panelist associated with the wearable device is determined to be in proximity to the television. In this case, the panelist is registered at the gateway **110** and an avatar for the panelist is displayed on the television. At this time, the registered panelist is associated with all identified content presented on the screen until the panelist is subsequently de-registered. On the other hand, if no confirmation signal is received from the wearable device, it is determined the panelist associated with the wearable device is not in proximity to the television, and the method moves on to step **1550**. At step **1550**, if the panelist is not currently registered at step **1550**, the method returns to step **1520** and simply continues to poll for Bluetooth communications from wearable electronic devices. However, if the panelist is currently registered at step **1550**, the method continues to step **1560** where the panelist is de-registered at the gateway **110**, and no avatar for the user is shown on the television. From this moment, the previously registered user will no longer be associated with any identified content presented on the television until the panelist is re-registered.

In view of the foregoing described processes of active and passive user registration, it will be recognized that registration of panelists may occur in any of several forms, including passive registration only, active registration only, or some combination of passive and active registration. In general, active registration and de-registration is only necessary in cases where a panelist is not carrying his or her smartphone or other mobile electronic device (e.g., wearable device), or if the mobile electronic device has run out of

battery power. Passive registration and de-registration occurs automatically when the user is carrying a fully powered mobile electronic device. In some embodiments, a warning message is shown prior to automatic de-registration of a panelist (e.g., "It appears that Adam has left the room; please press Adam's registration button on the remote control if this is incorrect.") These warning messages are designed to guard against de-registration when the user is actually still in the room but appears to have left the room for some reason (e.g., a panelist may appear to have left a room because his or her phone died, a child or other third party carried the user's smartphone out of the room, the user turned off the Wi-Fi on the phone, etc.). In further embodiments, when the system **100** detects that active and passive registration are often inconsistent, a message may be displayed on the television asking the panelist to take care to comply with registration protocols. For example, if a panelist repeatedly performs the active de-registration procedure but leaves his or her smartphone in the same room as the gateway **110**, a message may be sent asking the panelist to keep the phone on his or her person, or find a different charging station for the phone.

While exemplary methodologies for active and passive registration and de-registration are described herein, it will be recognized that various additional devices and methodologies may be used in addition to or in lieu of those described herein to confirm registration or de-registration of a panelist. For example, in at least one alternative embodiment, the gateway **110** is further equipped with a camera and is able to perform facial recognition on individuals within proximity of the gateway.

HDMI Overlay

The gateway **110** is equipped with HDMI overlay capabilities that allow content generated by the gateway **110** to be overlaid on content from the input source and displayed on the television **200**. The HDMI overlay capabilities are provided by the content overlay engine **168** (see FIG. **2A**), which is configured to overlay legends, messages, icons, avatars, and other additional content/information on the media content provided to the television via the cable connected to the HDMI out port **133**. The HDMI overlay engine **168** may be configured to overlay different types of information on the television at different times during operation of the gateway.

One instance in which the content overlay engine **168** overlays information on the media content is during panelist registration and de-registration. As described above, a panelist who is associated in the gateway **110** with a properly configured mobile electronic device (e.g., a smartphone or wearable electronic device) will be registered when the gateway detects that the mobile electronic device is in proximity to the gateway **110**. Each panelist in the household is associated with a unique avatar. When a panelist is registered, the gateway **110** displays the panelist's avatar over the content presented on the television for some period of time along with a welcome message to the newly registered panelist. When the panelist is de-registered, the gateway **110** displays a good-bye message and removes the panelist's avatar from the screen.

FIG. **16** shows an exemplary television **200** with a plurality of avatars **270** overlaid on the program content **254** presented on the television screen. Six avatars **270** are overlaid on the program content **254** in the example of FIG. **16**, indicating that six panelists are currently registered and are being associated with the program content **254** within the gateway **110**. The six avatars **270** are all included in a single row across the top left side of the television **200**. Avatars **270**

US 10,932,002 B2

41

for the currently registered panelists are all temporarily displayed on the television for a short period of time when a panelist registers or de-registers with the gateway 110. For example, the avatars 270 may be shown for five to ten seconds after an additional panelist registers with the gateway 110. After the short period of time, the avatars 270 disappear and only the program content 254 is shown on the television screen. Alternatively, in at least one embodiment, the avatars 270 are displayed in full color during the short period of time, but after expiration of the short period of time the avatars are muted (e.g., shown as dim colors, translucent, minimized, ghosted, replaced with small substitute icons) or completely hidden.

In the example of FIG. 16, the Jerry Smith recently entered the room where the television 200 is located carrying his smartphone. At this time, the power signal level associated with the MAC address for his smartphone exceeded the predetermined threshold and Jerry Smith was automatically registered with the gateway 110. This new registration prompted a welcome message 274 on the television (i.e., "Welcome Jerry"). Jerry Smith's avatar 272 was then displayed at the top of the television with the other avatars 270 for the currently registered panelists. Thereafter, during Jerry Smith's viewing session, all captured/identified content shown on the television 200 will be associated with Jerry Smith as well as any other registered panelists at the time of content identification (i.e., the data identifying the content will have the registered panelist's id and current timestamp attached to it).

After displaying the avatars 270 and welcome message 274 for a short period of time (e.g., five seconds), the welcome message 274 disappears, and the avatars 270 are muted on top of the television. For example, as shown in FIG. 17, the avatars 270 are replaced on the screen by small icons/shapes 271 (e.g., small circles) that are each personal to one of the panelists. These icons/shapes are significantly smaller than the avatars 270 and are intended to not be overly invasive on top of the media content 254 displayed on the screen. When the icons are all the same shape, each panelist may be associated with a different color (e.g., Jerry Smith is blue and Julie Smith is red) such that the panelists may quickly identify their personal registration icon on the screen.

Following registration of a panelist, the gateway continually scans for signals associated with panelists in order to determine panelist proximity to the television (i.e., by monitoring for signal strengths in excess of the threshold from any of various wireless devices associated with the panelist). If the panelist's mobile electronic device remains in proximity to the gateway 110 (i.e., the signal strength remains in excess of the threshold), the user will continue to be registered. However, when the gateway 110 does not detect the user's mobile device in proximity to the gateway, the user is de-registered and a good-bye message (e.g., "Goodbye Jerry Smith") is displayed on the television for some period of time (e.g., 5 seconds). At this time, the user's ID and timestamp will no longer be attached to the detected/captured content information.

While the foregoing discussion related to avatars 270 and related content overlaid on the television screen was discussed in the context of passive panelist registration and de-registration, it will be recognized that avatars 270 are similarly displayed during active panelist registration and de-registration. For example, if a user enters the room without any mobile electronic device, the user may actively register by simply pressing the active registration button on the remote control and his or her avatar will be overlaid on

42

the screen with a welcome message. In at least one embodiment, when a panelist passively registers or de-registers, a message is overlaid on the screen requesting confirmation of such registration or de-registration with the remote control. For example, if the signal strength associated with a registered panelist's mobile device is lost or is less than a threshold, a message is overlaid on the television asking the remaining panelists to actively de-register the departing panelist (e.g., "If Jerry Smith is no longer watching, please de-register him using the remote control."). In this manner, active means are used to control which panelists are registered, but passive means are used to prompt active registration or deregistration.

In yet another embodiment the gateway 110 is configured to periodically request panelist registration status. This periodic confirmation request may occur when the same panelists have all been registered for a long period of time (e.g., more than an hour). For example, as shown in FIG. 17, a message banner 276 is shown at the bottom of the screen asking "Is everyone still watching TV." This prompts the current viewers to look at the avatars 270 or registration icons 271 shown on the screen and make sure that all the registered panelists are still in the room. The remote control may then be used to actively de-register any panelists who are no longer in the room.

In view of the foregoing, it will be recognized that the gateway 110 is configured to overlay different legends, specific messages, or any of various other additional content directly over existing media content on the television 200. This is accomplished without the need of secondary screens or smaller displays that are hard to see. By using the television 200 as the display interface, the gateway 110 is capable of conveniently displaying any of various graphics, messages, and high-quality images to the users. As a result, the gateway 110 is equipped with numerous features that make the device highly functional and user-friendly.

Internet Activity Measurement

In addition to identifying media content presented on the television 200, the gateway 110 is also configured to identify media content presented on any of various wireless devices within the household. To this end, the gateway 110 is configured to act as a Wi-Fi router or sniffer. Operational software for wireless networking features is retained in the memory 116 of the gateway. The wireless networking software may be retained in a separate memory of the communications module 120, or may be retained with other instructional programs in the main memory of the gateway. In any event, this wireless software interacts with the networking hardware components (e.g., the Wi-Fi chipset 125) in order to provide routing services, and at the same time perform network sniffing that allows the gateway to detect each panelist's Internet activity on their associated mobile device and/or computer. The gateway 110 has two different operation modes that allow it to identify and measure the online mobile device/computer activity in the households: a router mode and a Wi-Fi Sniffer mode.

When operating in the router mode, the Wi-Fi chipset 125 allows the gateway 110 to operate as wireless access point or wireless signal repeater for the household 202. When operating in this mode, mobile/desktop clients connect directly to the gateway 110 in order to obtain an Internet connection. Thus, all the wireless traffic for the household 202 goes through the gateway 110. The gateway 110 is configured to capture the network packets, identify media content presented on specific devices, and generally log all the Internet traffic passing through the gateway.

With reference now to FIG. 18, an exemplary log 1800 is shown of network traffic captured by the gateway 100 when operating in the router mode. As shown in FIG. 18, the log 1800 includes a list of data packets 1810 transferred to various mobile devices via the gateway. Each data packet 1810 includes and/or is further associated with the following information at the gateway 110: date, time, MAC address of mobile device (or computer device), origin IP address, destination IP address, consumed URL, time to live, and user agent. This data is then stored in the memory of the gateway 110, and/or transferred to the remote server 310 for further processing and storage.

FIG. 19 shows an exemplary flowchart of a method 1900 of capturing network traffic at the gateway 110 operating in the router mode. The method 1900 begins when the gateway 110 is turned on at step 1910. At step 1920, the Wi-Fi chipset 125 begins operation in the route mode and the gateway 110 broadcasts the network name (SSID) to all wireless computing devices (including mobile devices and standalone devices) within the household 202. At step 1930, devices within the household are joined to the wireless network provided by the gateway 110. When each wireless device is initially joined to the network, the gateway 110 asks the user to associate one of the panelists within the household with the device. If a panelist is identified, the MAC address for the device is associated with the identified panelist within the gateway 110. If no panelist is associated with the device, the device is simply considered a generic traffic device. As shown in step 1940, the gateway 110 serves as the wireless access point for the wireless devices within the household, and routinely captures web traffic data, including web traffic data identifying all media content presented at the each MAC address. At step 1950, each time web traffic is captured, a determination is made whether the traffic is associated with one of the panelist's devices. If the web traffic is associated with a panelist's device, the method continues at step 1960 and associates the identified web traffic with the panelist. On the other hand, if the web traffic is not associated with a panelist's device, the method continues at step 1970, and the web traffic is associated to a generic network device (e.g., guest device). At step 1980, all the web traffic collected by the gateway 110 is compressed prior to transmission to the remote server 310. Thereafter, at step 1990, all the filtered and compressed traffic data is sent to the remote server(s) 310 for further analysis. In particular, if the gateway 110 does not include sufficient processing power to identify media content from the network traffic, such processing occurs at the more powerful remote server 310.

In addition to the router mode, the gateway 110 is also configured to operate in the Wi-Fi sniffer mode (which may also be referred to as the "promiscuous mode"). In this mode, the gateway 110 does not serve as a router, but instead join the household's existing wireless network (e.g., provided by the ISP's router within the household). After joining the household's existing wireless network, the gateway 110 then operates in the promiscuous mode and sniffs network packets that are passed through the network between various wireless devices and the network router. In general, the promiscuous mode causes the gateway 110 to pass all traffic/frames it receives to its microprocessor 114 (i.e., including traffic intended for other devices) for further processing, rather than passing only the traffic/frames specifically intended for the gateway 110. In this manner the gateway 110 analyzes all network traffic, and not only the traffic intended for the gateway 110. Data collected in the promiscuous mode includes the following for each data

packet: SSID, BSSID, signal strength/power, beacons, data, channel, encryption type, authentication type, and URL.

FIG. 20 shows an exemplary flowchart of a method 2000 of capturing network traffic at the gateway 110 operating in the promiscuous mode. The method 2000 begins when the gateway 110 is turned on at step 2010. At step 2020, the gateway 110 joins the household's Wi-Fi network and the Wi-Fi chipset 125 begins operation in the promiscuous mode. At step 2030, the Wi-Fi sniffer process runs, and the gateway looks for all traffic on the household's Wi-Fi network, including traffic not intended for the gateway 110. At step 2040, the gateway captures network traffic, and particularly URLs, consumed by/delivered to other network devices. At step 2050, each time web traffic is captured, a determination is made whether the traffic is associated with one of the panelist's devices. If the web traffic is associated with a panelist's device, the method continues at step 2060 and associates the identified web traffic with the panelist. On the other hand, if the web traffic is not associated with a panelist's device, the method continues at step 2070, and the web traffic is associated to a generic network device (e.g., guest device). At step 2080, all the web traffic collected by the gateway 110 is compressed prior to transmission to the remote server 310. Thereafter, at step 2090, all the filtered and compressed traffic data is sent to the remote server(s) 310 (e.g., in the cloud) for further analysis. In particular, if the gateway 110 does not include sufficient processing power to identify media content from the network traffic, such processing occurs at the more powerful remote server 310.

The foregoing process of collecting an identifying media content presented at various wireless devices is run in parallel with the other gateway processes, and in particular the process of identifying content presented on the television (e.g., see FIGS. 5A-5D and 8-10K) and the associated process of panelist registration (e.g., see FIGS. 12-17). With these processes all run in parallel on the gateway 110, it will be recognized that the gateway 110 is configured to identify content presented on all or nearly all media devices within the home, including one or more televisions 200 (with a gateway connected thereto), and any number of different wireless devices within the household. This capability allows the gateway 110 to serve as a single source capable of identifying all media content consumed within the household. Media content identified by the gateway 110 is conveniently summarized in data packages, as described below, and transmitted to the remote server.

Data Packages

FIG. 21A shows a schematic diagram of exemplary data packages 180 generated by the gateway 110. As shown in FIG. 21A, the exemplary data packages include each of the following: content data packages 182, presence data packages 184, demographic data packages 186, and system data packages 188. One or more of these data packages are automatically generated following a trigger event wherein a video frame is analyzed in-depth and the associated content identified. In particular, at least a content data package and a presence data package 184 are generated following a trigger event. These data packages may be immediately transferred to the remote server 310 and/or the cloud, or may be temporarily stored on the gateway 110 for subsequent transfer. The data packages 180 may be combined or otherwise associated with each other during transmission from the gateway 110. For example, a content data package 180 (i.e., identifying media content) and a presence data package 182 (i.e., identifying registered panelists associated with the identified content) may be automatically combined and

US 10,932,002 B2

45

transmitted following a trigger event. The remote server 310 may periodically request data from the gateway, such as a system data package 186 that provides diagnostic information about the gateway 110.

The content data packages 182 and presence data packages 184 transmitted to the remote server 310 are considered to be audience measurement “raw data”. This data is further analyzed with additional processing engines at the remote server 310 (or any number of additional remote/cloud servers). The remote server 310 (or servers) apply the appropriate editing rules in order to structure the data as “clean” data for final consumption by clients. “Clean” data may simply be data that is standardized in some manner. For example, is the content data from one cable provider states a program name of “Criminal Minds (2007)”, the remote editing rules may recognize that the “2007” parenthetical is a year of first showing, not the actual program name and may standardize the program name so simply be “Criminal Minds” so that it is consistent with the data collected from other cable providers. An exemplary database 190 including records/data packages of clean data is shown in FIGS. 21A and 21B. As noted in the figures, exemplary fields for the data package include viewership identification data (i.e., panelist identification), household identification data, content data, media source data, content provider, network data, channel data, program data, viewership room data (i.e., the room in the household where the content was viewed) viewership file, television provider, and any number of additional fields of data collected by the system 100.

Once the gateway 110 and/or remote server 310 has finished filtering, the resulting compliant (i.e., clean) data is stored by in a separate database. This data is prepared for generating reports and be consulted through a data provisioning portal and API. Final, clean data is available to various end game customers through an authenticated API that they can connect to their own systems for further processing, such as weighting, reporting or business intelligence systems. The aforementioned distributed processing/networking approach, wherein some of the analysis and machine learning routines are performed at the gateway 110 and additional machine learning is performed at the remote server, is advantageous. In particular, much of the data can be processed without the need of the transmission of the media files over the internet, thus saving costs of bandwidth and cloud infrastructure usage. At the same time, advanced processing at the remote server 310 means that the gateway 110 does not need the same advanced processing capabilities, thus resulting in various savings with respect to cost of each gateway.

The data generated and transmitted by the cross-media measurement system 100 is integrated with Blockchain. The use of Blockchain provides a reliable mechanism for audit automation and validation. By incorporating Blockchain in all the layers of the media measurement process, the system can ensure that all registrations, transactions and data generation are traceable and secure. In its simplest form, a Blockchain can be considered to be a distributed ledger which contains the relevant details for every transaction that has ever been processed. The validity and authenticity of each transaction is protected by digital signatures (cryptography). With Blockchain, there is no central administration, and anyone can process transactions using the computing power of specialized hardware. By using Blockchain, a distributed, cryptographic and immutable database is created. The database is considered to be distributed because, unlike most databases that control who can access the information in a system, any computer in the system can

46

access the Blockchain. This creates a system of trust since there is no centralized data. The database is considered to be cryptographic because every transaction recorded in the system is cryptographically verified to ensure its authenticity. Cryptography allows the system’s components to collaborate in an automated system of mathematical trust. The database is considered to be immutable because no records can be changed or altered; only new records can be appended to the distributed database. This ensures that data cannot be modified or altered in a way that would change the data generated by the system 100.

Remote Processing

As discussed above, the system 100 is configured to process data captured by the gateway 110 at either the gateway itself, or at one or more remote computing devices, such as remote server 310. The remote server 310 is merely representative of any number of remote computing devices and/or cloud based software that may be utilized by the system 100. The use of remote computing devices and cloud based software allows for increased processing power, expanded memory, and overall increased system functionality.

One example of additional functionality with cloud-based software is increase machine learning processing. In at least one embodiment, all software functions associated with content recognition and processing may run directly in the cloud, and not on physical server or even virtual servers. In this embodiment, the content recognition features work on-demand, and every time a new media capture arrives to the cloud storage, these functions automatically execute to process each piece of media individually. The training/machine learning models that power the cloud-based recognition are much more robust than the ones running locally in the gateway 110. Although the output data is generally the same in structure, the level of training and accuracy of these models may be more advanced since the computing power in the cloud is higher than that running in the gateway itself.

In addition to machine learning, the remote processing capabilities of the system 100 also include raw data storage and processing. The remote software interprets the data generated from the machine learning process and stores all the found metadata into a high-performance big-data database. This data has not yet been processed with editing rules; it is stored directly as it arrives from the machine learning process.

The remote processing power of the system further facilitates the application of compliance rules to the data collected from the gateway 110. Compliance rules are needed for any audience measurement process. These rules are applied to the raw data, in order to determine which households and household members are compliant and adequate to participate in the measurement for a particular day. In the present system, the compliance rules include (i) filtering of households that have a problem with the device or have disconnected it (i.e., diagnostic routines report trouble with the gateway), (ii) filtering of households that have more televisions than gateways 110 installed (i.e., all televisions of the household must be measured in order to accurately measure media consumption within the household), and (iii) filtering of household members that have more than some predetermined threshold period (e.g., 24 hours) of continuous TV viewing (i.e., thus indicating that the panelist is not actually watching the content presented). Once the processing module (e.g., at remote server 310) has finished filtering, the resulting compliant (clean) data is stored by this module in

US 10,932,002 B2

47

a separate database. This data is prepared for generating reports and be consulted through the data provisioning portal and API.

In at least one embodiment, the remote server 310 is configured to provide a web-based piece of software that allows clients to access the collected data and visualize in on a GUI. The GUI may include a dashboard visualization where all the viewing session of a particular household is displayed. This may include the person recognition information (times where a user was watching television), and a timeline that is constructed based on the time a user spent watching a specific platform, source, channel, network or program. FIG. 6 serves as an example of an exemplary dashboard for a GUI, but it will be recognized that numerous other dashboards are also possible.

In at least one embodiment, a data API is provided for clients via the remote server 310. Clients are able to “pull” the information collected from various gateways 110 directly from the API in order to generate their own reports or to connect our data to existing systems. User administration, permission control and setup are performed by an in-house team.

Television ON/OFF State Detection

It will be appreciated based on the foregoing description that the gateway 110 is configured to identify content consumed by panelists across numerous devices, including content delivered to wireless devices 220 (e.g., smartphones, tablets, etc.) and content delivered to the television 200. The gateway 110 is configured to continually monitor and identify content delivered to the mobile devices 110. However, the gateway 110 is only configured to monitor and identify content delivered to the television 200 when the television itself is turned on (i.e., such that the television screen and capable of presenting content to panelists within the household). By limiting content identification to times when the television is actually turned on, content consumption is more accurately determined. Furthermore, limiting content identification times further optimizes the use of computing resources within the gateway 110 and saves data consumption associated with data transmission through the gateway’s Wi-Fi or cellular network connections.

The gateway 110 is configured to determine the on/off state of the television in two ways. First, the gateway 110 is configured to determine the on/off state of the television 200 by monitoring the consumer electronics control (CEC) pin on the HDMI connection between the television 200 and the gateway 110 (i.e., at the HDMI OUT port). Second, the gateway is configured to determine the on/off state of the television 200 by monitoring power flowing to the television 200 via the gateway 110.

AC Detection Through HDMI-CEC

CEC is a control function that lets one A/V component control another if they are connected via HDMI cables. If the television 200 is CEC-enabled, power detection can be performed through the HDMI cable connected to the HDMI OUT port 133 of the gateway 110. By monitoring the CEC pin of the HDMI cable, the gateway 110 can detect signals indicating that the television has been turned on or turned off. As noted previously, in at least some embodiments the gateway 110 is configured to save energy and stop performing television audience measurement computing processes (including content identification and panelist registration) when the television is off. Additionally, in at least some embodiments, the gateway 110 is configured to power itself down whenever the television 200 is turned off. This is especially true in embodiments where the gateway 110 is not

48

monitoring mobile device traffic, and is only monitoring content consumed at the television.

FIG. 22 shows a flowchart 2100 of a process used by the gateway 110 for determining television state based on CEC. The process begins with step 2110 when the gateway 110 is turned on. Thereafter, as step 2120, the CEC pin at the HDMI output port 133 is monitored to detect whether the television is on or off. At step 2130, a signal is received at the CEC pin. If the CEC pin indicates that the television is on, the AC status in the gateway 110 is set to “1” at step 2140. If the CEC pin indicates that the television is off, the AC status in the gateway 110 is set to “0” at step 2150. At step 2160, the television status is made available for output to other components and devices via an application programming interface (API). The process ends at step 2170, but the process is periodically repeated starting with step 2120 in order to determine television status based on the HDMI-CEC port.

AC Detection Through Gateway Power Detection

In addition to being configured to determine power via the CEC control function, the gateway 110 is also configured to monitor power delivered to the television via the AC output port 144. To this end, the AC output port 144 of the gateway 110 includes a receptacle that receives the television’s AC plug. When the television plug is connected to the AC output port 144 of the gateway 110, the gateway is able to determine whether power is flowing to the television.

As discussed previously in association with FIG. 2B, the gateway 110 includes an integrated power supply 150 that powers all the electronic components inside of the housing 112. The gateway 110 also includes a TV ON/OFF detection circuit 151 that is capable of sensing that alternating current (AC) is flowing to the television’s power cable through the AC output port 144. Through an embedded API, the gateway 110 software obtains the readings from the TV ON/OFF detection circuit 151 to detect if the TV Set is turned on (i.e., alternating current is flowing to the television in excess of a threshold) or is turned off (no/low alternating current to the television).

FIG. 23 shows a flowchart 2200 of a CoreMeter process for determining television state based on AC detection using an AC sensor provided by the TV ON/OFF detection circuit 151. The process begins at step 2210 when the gateway is turned on. Then at step 2220, the TV ON/OFF detection circuit 151 of the gateway 110 monitors AC power flowing to the television 200 (e.g., via a current sensor). If AC power is detected at step 2230, the process moves to step 2240 and a determination is made that the current provided to the television is within a calibration threshold. If the current is within the threshold, the process moves to step 2250, and the gateway status of AC to the television is set to “1”. On the other hand, if no AC current is detected at step 2230, or if the AC current detected is not within the predetermined threshold at step 2240, the process moves to step 2260, and the gateway status of AC to the television is set to “0”. At step 2270, the television status is made available for output to other components and devices via an application programming interface (API). The process ends at step 2280, but the process is periodically repeated starting with step 2220 in order to determine television status based on the AC to the television.

Advantageously, the two different mechanisms for monitoring power to the television 200 (i.e., CEC detection and AC detection) allows the gateway 110 to provide diagnostic information/warnings to the household when there are issues with the connections between the gateway and the television. As a first example, if the power cord of the television

200 is improperly plugged into an AC wall outlet instead of the AC output port 144 of the gateway 110, but the CEC pin on the HDMI OUT port 133 indicates that the television 200 was recently turned on, the gateway 110 may send a warning message for display on the television 200 instructing the user to plug the television into the AC output port 144 of the gateway 110. As another example, if the power cord of the television 200 is properly plugged into the AC output port 144 of the gateway 110, but there is no signal at the CEC pin on the HDMI OUT port 133, this may indicate that the HDMI connection between the television 200 and the gateway 110 has become disconnected, and a warning message can be delivered to the user to check the HDMI connection (e.g., a message delivered via the LCD display 152 or microphone 154 of the gateway).

Graphical User Interface

The main user interface of the gateway is user-friendly, attractive, and generally makes use of all media sources easy and convenient. This encourages panelists to consume media content via the gateway, thus allowing the gateway 110 to perform the measurement functions. There are at least two options for the gateway's main user interface. FIG. 24 shows a first option for the gateway main user interface wherein the currently selected media source (e.g., TV, game console, etc.) plays in the background, HDMI sources 210 are provided along an arc on a left side of the display, and OTT sources 161 are provided along an arc on a right side of the display. FIG. 25 shows a second option for the gateway main user interface wherein the currently selected media source plays in the background, HDMI sources 210 are provided on a top row of the display, and OTT sources 161 are provided along lower rows of the display.

Exemplary Technological Improvements

Based on the foregoing description, it will be recognized that the system and method for cross-media measurement described herein provides a technological improvement in the form of improved hardware and software devices for reviewing, analyzing and capturing media content presented on a television and various additional media presentation devices within a household. The system provides specific improvements of the conventional systems and related methods. Examples of these improvements over conventional ACR systems are included below. However, the improvements over conventional systems and method are not limited to the examples provided below.

As discussed previously, ACR technology is based on the use of an audio fingerprinting/matching technology, where binary files are compared to a series of audio/pixel references to determine which channels are being watched. In contrast to ACR technology, the gateway 110 leverages machine learning technologies (e.g., computer vision) to perform channel, content and advertising detection directly from the incoming video source, from its embedded OTA Tuner, and from the network traffic coming in/out connected devices. The gateway does not take a single approach for all media consumption options. Instead, the gateway considers the various methods of viewing media, and takes a multi-layer approach to determine the content presented to panelists. The methodology of the gateway does not need to assume the source of content, but instead bases content analysis on the source. This provides for more accurate content recognition and analysis. Furthermore, the gateway is able to definitively identify the decision-making path of content selection.

In order to comprehensively support an ACR solution, the library used for its implementation must include continuous recording of all available TV signals. Recorded stations and

geographically distributed backups must be installed at different locations nationwide. The libraries should further include original content that is available across every available streaming service. Advantageously, the gateway 110 does not rely on a reference library. Instead, the gateway makes use of trained machine learning models in order to detect particular items in the watched signal. This is significant both from an accuracy perspective as well as a comprehensive analysis perspective. The gateway is able to reliably detect and measure content that may not be included in an ACR library.

ACR is completely unable to detect a channel/source provider when an event or program is being transmitted simultaneously in different channels. In contrast, the gateway does not have simulcast detection problems since it does not rely on audio. The gateway explicitly tracks the specific source of the content and ads that are delivered to the television. With this methodology we are also able to measure simulcast events (same content broadcasted at the same time in different channels), which is not possible with the ACR approach. The ability to identify sources allows for accurate reporting across the various television platform and device options which then translates to accurate allocation of measurement for both content and ads.

With ACR, beyond schedule supported content environments (i.e., viewing through the use of additional external devices such as video game consoles, DVD players, OTT sources, etc.) are difficult or impossible to detect. The conventional solution to this is a content matching methodology wherein validation of exposure/source must occur via manual panelist confirmation (e.g., manual button pushing). In contrast, the gateway is configured to detect and measure content from any number of different sources (e.g., video games, cable boxes, Blu-ray players, OTT sources, etc.). Thus, the gateway is configured to determine viewing behavior outside of non-linear environments (AppleTV, Roku, Video Game Console, etc.), inclusive of source, content, and ads. Marketers need validated measurement of these environments in order to shift their ad dollars into them.

In addition to the above, conventional ACR systems are not capable of measuring content presented at secondary devices (e.g., mobile phones, computers, etc.). Accordingly, additional hardware, software and meter components are needed in order to measure secondary devices in the ACR environment. In contrast, the gateway includes a dedicated chipset that allows it to act as wireless access point to capture the traffic going through it. This allows the entire household's media traffic to be analyzed and reported by the gateway. The gateway thus provides a single-source cross-platform solution to audience measurement that is capable of detecting media consumption overlaps across television platforms and secondary devices.

With conventional ACR, the timeliness of reporting is reliant on the ACR match-back process which needs at least 24 hours for the various confirmations needed to make the inferences of measurement. In the event of recording quality issues, reprocessing has to be re-run, and information delivery can be importantly delayed or entirely left out. The gateway 110 delivers content detection as it happens, and the information can be delivered to the cloud essentially in real time.

Conventional ACR and related systems require additional peripheral hardware to perform TV ON/OFF detection. In particular, the use of ACR microphones to detect TV ON/OFF is unreliable as the microphones tend to pick up signals from other sources which skews the output data (e.g.,

US 10,932,002 B2

51

the microphone may detect television on when the audio is actually from a radio). In contrast to conventional ACR, the configuration of the gateway 110 allows it to definitively identify TV ON/OFF without reliance on a microphone or any peripheral device. All hardware required for TV ON/OFF detection is included inside the gateway 110, thus allowing TV ON/OFF detection as a built-in feature. As described above, in various embodiments, an AC Loop circuit detects the power consumed by the television connected to the gateway. Additionally, for those televisions that support the HDMI CEC protocol, the detection of TV ON/OFF state can be done via the HDMI output port of the gateway.

Conventional ACR is dependent on the capture of very high quality audio via microphones external to the television. This is problematic and unreliable for numerous reasons, including the possibility for muted televisions, ambient noise, etc. In contrast, the gateway 110 is configured to perform recognition through video content analysis, and particularly analysis of the video signal itself (e.g., in the case of an OTA television signal) or the selected frames of an input source (e.g., in the case of content provided by a cable box). Video analysis (e.g., video frame analysis) is more accurate and effective than audio detection because video analysis avoid the problems associated with audio capture including probability of interference and lack accurate identification and detection.

ACR supported methodologies utilize consumer-grade off the shelf hardware (e.g., portable tablets) which is not meant for the panel environment (thus resulting in short shelf lives). ACR supported methodologies also cannot support a variable device multimedia entertainment system. A patched together approach of several devices is usually necessary and implemented. In contrast to ACR, the gateway is a self-contained system wherein every component required to support each layer of the measurement solution is soldered to the main board. No delicate parts can be easily broken which results in long shelf lives that support the panel environment. The hardware is specifically designed for the in-home panel environment. It is configured to analyze and detect numerous different methods of media consumption. The gateway allows for a passive panelist experience which results in the most accurate, comprehensive and granular single-source data output.

In addition to the above, the consumer-grade off the shelf hardware components and devices (e.g., portable tablet computers) common to many ACR systems are attractive for panelists to use for tasks outside of panel measurement objectives. These devices are thus subject to abuse and a generally shorter lifespan. These devices must also be repeatedly turned on and off and charged, such that a relatively short lifespan is inherent with the device. In contrast, the minimalist design of the gateway 110 results in an innocuous presence in the household that discourages tampering. This reduces the opportunities for equipment loss due to fraud and allows for consistent presence within the household for audience measurement purposes. The gateway is specifically designed for continuous audience measurement (i.e., 24 hours a day, seven days a week) without the need for the user to take any particular action.

Because ACR technologies implement consumer-grade off-the-shelf hardware devices, user experiences are often forced upon the panelist experience when interacting with these devices that would otherwise not be in the home. In contrast, the gateway provides a friendly user interface that leverages the use of the television as a display to show high impact graphics. The panelist experience and behaviors are

52

consistent with those prior to our installation of the gateway. The ability to communicate on the television screen allows the user interface to remain native to the user's existing media consumption environment.

The aforementioned combination of several components, devices, and consumer-grade off-the-shelf hardware with conventional ACR systems has long-term cost implications. The multiple points of failure results in high equipment churn, replacement, and repair costs. This equipment churn also impacts panelist satisfaction/drop-out, which in-turn results in panelist replacement costs. In contrast, the gateway includes all the necessary hardware in one device, is extremely durable, is built for continuous audience measurement real-world environment, has an innocuous presence, and includes an all-in-one measurement approach. As a result, the gateway 110 not only delivers higher data integrity, but also facilitates unique data outputs that can produce higher return on investment.

Although the various embodiments have been provided herein, it will be appreciated by those of skill in the art that other implementations and adaptations are possible. Furthermore, aspects of the various embodiments described herein may be combined or substituted with aspects from other features to arrive at different embodiments from those described herein. Thus, it will be appreciated that various of the above-disclosed and other features and functions, or alternatives thereof, may be desirably combined into many other different systems or applications. Various presently unforeseen or unanticipated alternatives, modifications, variations, or improvements therein may be subsequently made by those skilled in the art which are also intended to be encompassed by the following claims.

What is claimed is:

1. A method of identifying media content presented on a display device in communication with a content gateway, the media content provided by a video signal comprising a series of frames, the method comprising:

determining, at a processor within the gateway, a selected input source providing the video signal, wherein the selected input source is one of a plurality of input sources including at least a first input source and a second input source;

selecting a first set of content identification rules when it is determined that the selected input source is the first input source, wherein the first set of content identification rules define a first trigger event and a first algorithm for analyzing one or more of the frames of the video signal following the first trigger event;

selecting a second set of content identification rules when it is determined that the selected input source is the second input source, wherein the second set of content identification rules define a second trigger event and a second algorithm for analyzing one or more frames of the video signal following the second trigger event, wherein the second set of content identification rules is different from the first set of content identification rules; and

applying the selected first set or second set of content identification rules to the video signal in order to generate content identification data for the media content presented on the display device, wherein applying the selected first set of content identification rules includes waiting for the first trigger event and applying the first algorithm to one or more frames of the video signal following the first trigger event, and wherein applying the selected second set of content identification rules includes waiting for the second trigger event

US 10,932,002 B2

53

and applying the second algorithm to one or more frames of the video signal following the second trigger event.

2. The method of claim 1 further comprising, transmitting the generated content identification data to a remote computing device.

3. The method of claim 1 wherein the first trigger event is detection of a content grid within one of the frames of the video signal, and wherein application of the first set of content identification rules includes periodically applying a grid detection algorithm to the frames of the video signal.

4. The method of claim 3 wherein the grid detection algorithm is applied to every frame of the video signal provided by the selected input source.

5. The method of claim 3 wherein the first algorithm is a content extraction algorithm applied to the detected content grid.

6. The method of claim 1 wherein the first input source is a specific model of set-top box, and wherein selection of the first identification rules is based at least in part on the specific model of set-top box.

7. The method of claim 1 wherein the second trigger event is passage of an amount of time since a previous frame capture, and wherein the second algorithm is a logo recognition algorithm.

8. The method of claim 1 wherein the second trigger event is detection of a scene change in the series of frames, and wherein the second algorithm is a brand recognition algorithm.

9. The method of claim 1 wherein the display device includes a screen and a speaker, wherein the first algorithm is a first machine-learned algorithm, and wherein the second algorithm is a second machine-learned algorithm.

10. A non-transitory computer-readable medium for identifying media content provided by a video signal delivered to and presented on a display device, the computer-readable medium having a plurality of instructions stored thereon that, when executed by a processor, cause the processor to:

determine a selected input source providing the video signal, wherein the selected input source is one of a plurality of input sources including at least a first input source and a second input source;

select a first set of content identification rules when it is determined that the selected input source is the first input source, wherein the first set of content identification rules define a first trigger event and a first algorithm for analyzing one or more frames of the video signal following the first trigger event;

select a second set of content identification rules when it is determined that the selected input source is the second input source, wherein the second set of content identification rules define a second trigger event and a second algorithm for analyzing one or more frames of the video signal following the second trigger event, wherein the second set of content identification rules is different from the first set of content identification rules; and

apply the selected first set or second set of content identification rules to the video signal in order to generate content identification data for the media content presented on the display device, wherein application of the selected first set of content identification rules causes the processor to wait for the first trigger event and apply the first algorithm to one or more frames of the video signal following the first trigger event, and wherein application of the selected second set of content identification rules causes the processor

54

to wait for the second trigger event and apply the second algorithm to one or more frames of the video signal following the second trigger event.

11. The non-transitory computer-readable medium of claim 9 wherein the first trigger event is detection of a content grid within one of the frames of the video signal, wherein application of the first set of content identification rules includes periodically applying a grid detection algorithm to the frames of the video signal, and wherein the first algorithm is a content extraction algorithm applied to the detected content grid.

12. The non-transitory computer-readable medium of claim 9 wherein the first input source is a specific model of set-top box, and wherein selection of the first identification rules is based at least in part on the specific model of set-top box.

13. The non-transitory computer-readable medium of claim 9 (i) wherein the second trigger event is one of passage of an amount of time since a previous frame capture or detection of a scene change in the series of frames, and (ii) wherein the second algorithm is a logo recognition algorithm or a brand recognition algorithm.

14. A gateway for identifying media content presented on a display device including a screen and a speaker, the gateway comprising:

a plurality of input ports including at least a first input port and a second input port;

an output port configured to transfer a video signal received at the first input port or the second input port to the display device, wherein the video signal includes a series of frames that provide the media content; and a processor configured to execute a computer application comprising a plurality of instructions which are configured to, when executed, cause the gateway to: determine a selected input port providing the video signal;

select a first set of content identification rules when it is determined that the selected input port is the first input port, wherein the first set of content identification rules define a first trigger event and a first algorithm for analyzing one or more frames of the video signal following the first trigger event;

select a second set of content identification rules when it is determined that the selected input port is the second input port, wherein the second set of content identification rules define a second trigger event and a second algorithm for analyzing one or more frames of the video signal following the second trigger event, wherein the second set of content identification rules is different from the first set of content identification rules; and

apply the selected first set or second set of content identification rules to the video signal in order to generate content identification data for the media content presented on the display device, wherein application of the selected first set of content identification rules causes the processor to wait for the first trigger event and apply the first algorithm to one or more frames of the video signal following the first trigger event, and wherein application of the selected second set of content identification rules causes the processor to wait for the second trigger event and apply the second algorithm to one or more frames of the video signal following the second trigger event.

15. The gateway of claim 14 further comprising a housing, wherein the processor, the plurality of input ports, and the output port are all retained within the housing, and

US 10,932,002 B2

55

wherein the plurality of input ports and the output port are accessible through the housing.

16. The gateway of claim 15 wherein the first input port and the second input port are both HDMI ports.

17. The gateway of claim 16 wherein the first input port 5 is connected to a set-top box, and wherein the second input port is connected to one of a video game console, a disc player, or an OTT device.

18. The gateway of claim 14 further comprising a storage apparatus and a transceiver, wherein the first set of content 10 identification rules and the second set of content identification rules are retained in the storage apparatus, wherein the transceiver is configured to communicate with a remote server, and wherein at least one of the first algorithm and the second algorithm is stored at the remote server, and wherein 15 application of the first application or the second application causes the processor to capture and transmit one or more frames of the series of frames to the remote server via the transceiver for application of the first algorithm or the second algorithm. 20

19. The gateway of claim 14 further comprising a router, wherein the processor is further configured to detect additional media content presented at a mobile device via the router.

20. The gateway of claim 15, wherein the processor is 25 configured to register a panelist for association with media presented on the display device based on a signal strength of the mobile device at the router.

* * * * *

56

EXHIBIT F

NEXT TV

Broadcasting + Cable
Multichannel News

Broadcasting+Cable



PREMION
Brand-Safe Premium CTV/OTT Advertising
TAG BRAND SAFETY CERTIFIED
[START MY CAMPAIGN](#)

TRENDING

New 'Hot Bench' Judges

The Death of Linear CPMs?

Holiday Programming

GAC App Debuts

[Home](#) > [News](#)

VideoAmp Incorporates HyphaMetrics Panel Data Into Audience Measurement

By [Jon Lafayette](#) ([Broadcasting & Cable](#)) published October 13, 2021

In big data world, panels still matter, companies say



VideoAmp said it made a deal that will enable it to incorporate data from [HyphaMetrics](#)' cross-screen audience panels into its measurement products.

BROADCASTING & CABLE
NEWSLETTER

Your Email Address

☐ I consent the use of my personal data in accordance with Future's [privacy policy](#), including its cookie policy.

SIGN ME UP

By submitting your information you agree to the [Terms & Conditions](#) and [Privacy Policy](#) and are aged 16 or over.

VideoAmp is one of a number of measurement companies jockeying to provide the TV industry with alternatives to Nielsen, which has been under fire because the way it managed its panels led to undercounting viewing during the pandemic.



Also Read: [Ad Industry Seeks Alternatives After Nielsen Loses Seal of Approval](#)

ViacomCBS said it would use a currency based on VideoAmp measurement as a currency to plan, transact and measure national media campaigns, a role monopolized by Nielsen.

Advertisement

The Kelley Drye logo consists of the word "Kelley" in a blue, sans-serif font and the word "Drye" in a green, sans-serif font, positioned directly below "Kelley".

**Blocked by
Palo Alto
Security**

**CAUTION: Your
requested URL has
been blocked by the
Palo Alto URL Filter
database module.**

**User Name: kdw\ben-
t**

**URL:
8e3b3c72bf948355482c9d98db538c
0-
40/html/container.html**

**Rule Name: GP-VPN
Internet URL Filter
default**

**Category: web-
advertisements**

Please click the link

Advertisement

"At VideoAmp we see panels as an important part of our overall solution set as the industry searches for a new holistic media measurement

currency," VideoAmp chief measurability officer Josh Chasin said. "We've been especially impressed with HyphaMetrics' innovative, patented measurement tech stack and we look forward to working together to develop new measurement solutions for buyers and sellers of cross-screen advertising."

VideoAmp said the way it is using HyphaMetrics panel-based data set aligns with the WFA's blueprint for cross-media measurement. The WFA outlines merging census-level big data and panel data to provide a better understanding of consumer media behavior.



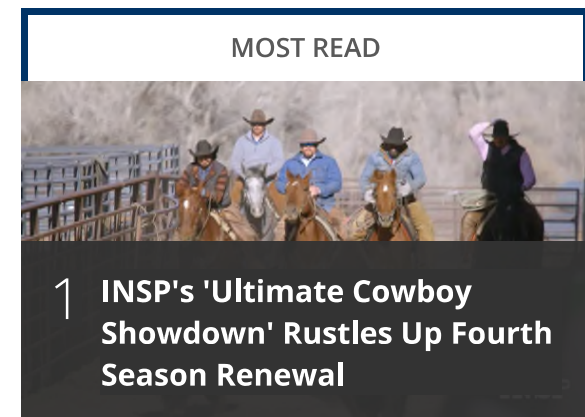
Joanna Drews
(Image credit:
HyphaMetrics)

"Panels play an integral role in providing a definitive understanding of how consumers interact with content and advertising across all of their devices and channels," said Joanna Drews, CEO and co-founder of HyphaMetrics. "We are excited for VideoAmp to incorporate this granular behavioral layer of data, covering all aspects of today's unique viewing environment such as video gaming, secondary device usage and walled gardens."

Drews, a former Comscore executive, started HyphaMetrics, which has developed its own meter to measure every network, program, advertisement, product placement, streaming app and gaming environment on every device in a household. The company expects to have 5,000 homes and 15,000 devices hooked up to its technology.

In March, [HyphaMetrics received a patent](#) for cross-platform measurement.

[Also Read: MadHive First To License Panel Data from HyphaMetrics](#)



VideoAmp uses a data set of co-mingled viewership from smart TVs and set-top boxes to measure video across devices. The data is housing on a platform where it can be utilized with tools for planning and optimizing campaigns and valuing media as its being bought and sold.

[Also Read: VideoAmp Renews Key Vizio Deal For TV Viewing Information](#)

VideoAmp is a charter member of HyphaMetrics Content Metrics panel trial. The trial, which commenced in July of 2021, will run through the first quarter of 2022 in order to encompass major viewing events like the Super Bowl and Winter Olympics.



Jon Lafayette



Jon has been business editor of Broadcasting+Cable since 2010. He focuses on revenue-generating activities, including advertising and distribution, as well as executive intrigue and merger and acquisition activity. Just about any story is fair game, if a dollar sign can make its way into the article. Before B+C, Jon covered the industry for TVWeek, Cable World, Electronic Media, Advertising Age and The New York Post. A native New Yorker, Jon is hiding in plain sight in the suburbs of Chicago.

MORE ABOUT BROADCASTING AND CABLE

LATEST



2 **Bounce Teams With MGM On Original Comedy 'Act Your Age'**

3 **Brodesser-Akner on Challenges of Taking 'Fleishman' From Book to TV**

4 **Al Pacino Drama 'Hunters' Has Second and Final Season on Prime Video January 13**

5 **Roku Cuts 200 Employees, Blames 'Current Economic Conditions'**

Advertisement

**Kelley
Drye**

**Blocked by
Palo Alto**



**Brodesser-Akner
on Challenges of
Taking 'Fleishman'
From Book to TV ►**



**Al Pacino Drama
'Hunters' Has
Second and Final
Season on Prime
Video January 13 ►**



**INSP's 'Ultimate
Cowboy
Showdown'
Rustles Up Fourth
Season Renewal ►**

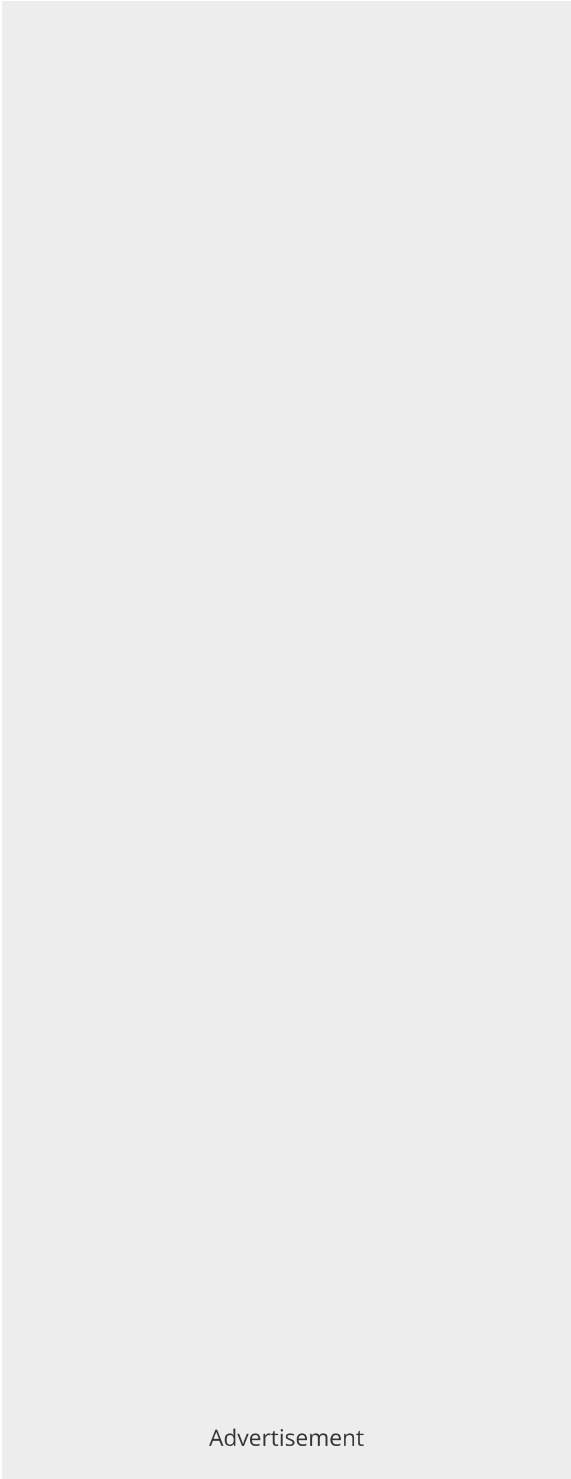
SEE MORE LATEST ►

Advertisement

ONE LA MEDIA
PARTNER.
EVERY SCREEN.


DO IT →

ADLINK
Powered by Spectrum Reach



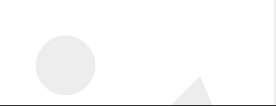
Advertisement

MOST POPULAR

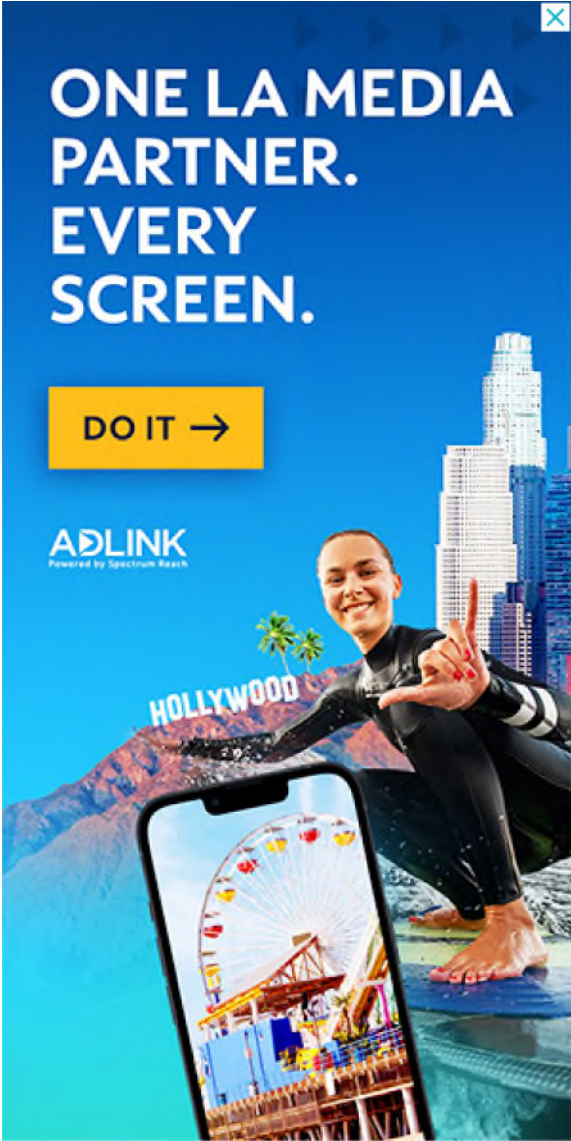


Fox Sports’s World Cup Coverage Plays on Broadcast and Cable Field

By [R. Thomas Umstead](#) about 2 hours ago



Fox Sports’s World Cup Coverage Plays on Broadcast and Cable Field



Faith jessie joins WXIA-TV in Atlanta As Weeknight Anchor

By [Jon Lafayette](#) about 3 hours ago

NewsNation Aims Billboard at CNN Staffers Facing Layoffs

By [Jon Lafayette](#) about 3 hours ago

FCC Unveils New Broadband Map

By [John Eggerton](#) about 4 hours ago

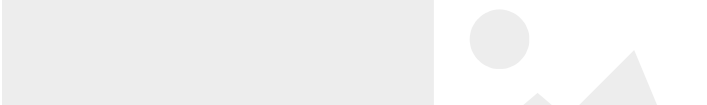
Advertisement



**GIVING
BIG BRANDS
LA FLAVOR**

DO IT →

ADLINK
Powered by Spectrum Reach



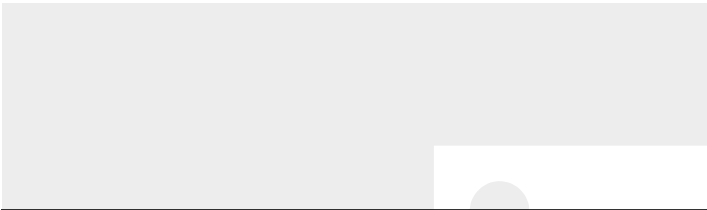
Bethenny Frankel Boards Kevin O'Leary's 'Money Court'

By [Michael Malone](#) about 5 hours ago



Hallmark Channel Sets January Premiere Date for 'The Way Home'

By [R. Thomas Umstead](#) about 15 hours ago



YouTube Leads All OTT Platforms, Including Netflix, in Monthly Average Viewers (Ridiculous Chart of the Day)

By [Daniel Frankel](#) about 19 hours ago



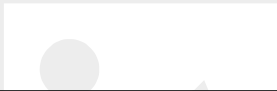
FCC Approves Broadband Label Rules

By [John Eggerton](#) about 21 hours ago



'America's Got Talent: All-Stars' Reveals Acts That Will Compete

By [Michael Malone](#) about 22 hours ago



Senate Democrats Seek FTC Investigation of Twitter

By [John Eggerton](#) about 22 hours ago

Broadcasting+Cable is part of Future US Inc, an international media group and leading digital publisher. **Visit our corporate site.**

[Terms and conditions](#)

[Privacy policy](#)

[Cookies policy](#)

[Accessibility statement](#)

[Careers](#)

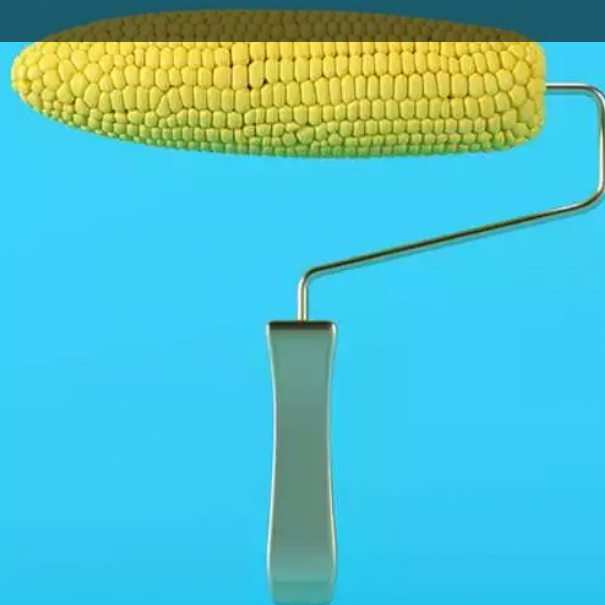
[Advertise with Us](#)

[Contact Us](#)

[Do not sell my info](#)

© Future US, Inc. Full 7th Floor, 130 West 42nd Street, New York, NY 10036.

EXHIBIT G



11 Dec, 2019

Image Recognition with Deep Neural Networks and its Use Cases

Reading time: 10 minutes

It's nice to realize that we humans managed to enable machines with our own natural skills: learning by example and perceiving the outside world. The only problem is that significantly more time and effort is needed to teach computers how to "see" like us. But if we think about the practical purpose this capability already brings to organizations and businesses, the effort pays off.

In this article, you'll learn what image recognition is and how it's related to computer vision. You'll also find out what neural networks are and how they learn to recognize what is depicted in images. Finally, we'll discuss some of the use cases for this technology across industries.



What is image recognition and computer vision?

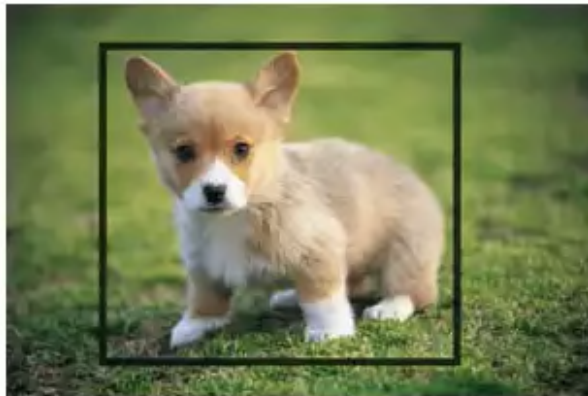
Image recognition (or image classification) is the task of identifying images and categorizing them in one of several predefined distinct classes. So, image recognition software and apps can define what's depicted in a picture and distinguish one object from another.

The field of study aimed at enabling machines with this ability is called **computer vision**. Being one of the computer vision (CV) tasks, image classification serves as the foundation for solving different CV problems, including:

Image classification with localization – placing an image in a given class and drawing a bounding box around an object to show where it's located in an image.



Object Classification is the task of identifying that picture is a dog



Object Localization involves the class label as well as a bounding box to show where the object is located.

Image classification vs image classification with localization. Source: [KDnuggets](#)

Object detection – categorizing multiple different objects in the image and showing the location of each of them with bounding boxes. So, it's a variation of the image classification with localization tasks for numerous objects.

Object (semantic) segmentation – identifying specific pixels belonging to each object in an image instead of drawing bounding boxes around each object as in object detection.



Instance segmentation – differentiating multiple objects (instances) belonging to the same class (each person in a group).



Object Detection



Semantic Segmentation



Instance Segmentation

The difference between object detection, semantic segmentation, and instance segmentation. Source: [Conditional Random Fields Meet Deep Neural Networks for Semantic Segmentation](#)

Researchers can use deep learning models for solving computer vision tasks. **Deep learning** is a machine learning technique that focuses on teaching machines to learn by example. Since most deep learning methods use neural network architectures, deep learning models are frequently called deep neural networks.

Deep neural networks: the “how” behind image recognition and other computer vision techniques

Image recognition is one of the tasks in which **deep neural networks** (DNNs) excel. Neural networks are computing systems designed to recognize patterns. Their architecture is inspired by the human brain structure, hence the name. They consist of three types of layers: input, hidden layers, and output. The input layer receives a signal, the hidden layer processes it, and the output layer makes a decision or a forecast about the input data. Each network layer consists of interconnected *nodes* (*artificial neurons*) that do the computation.

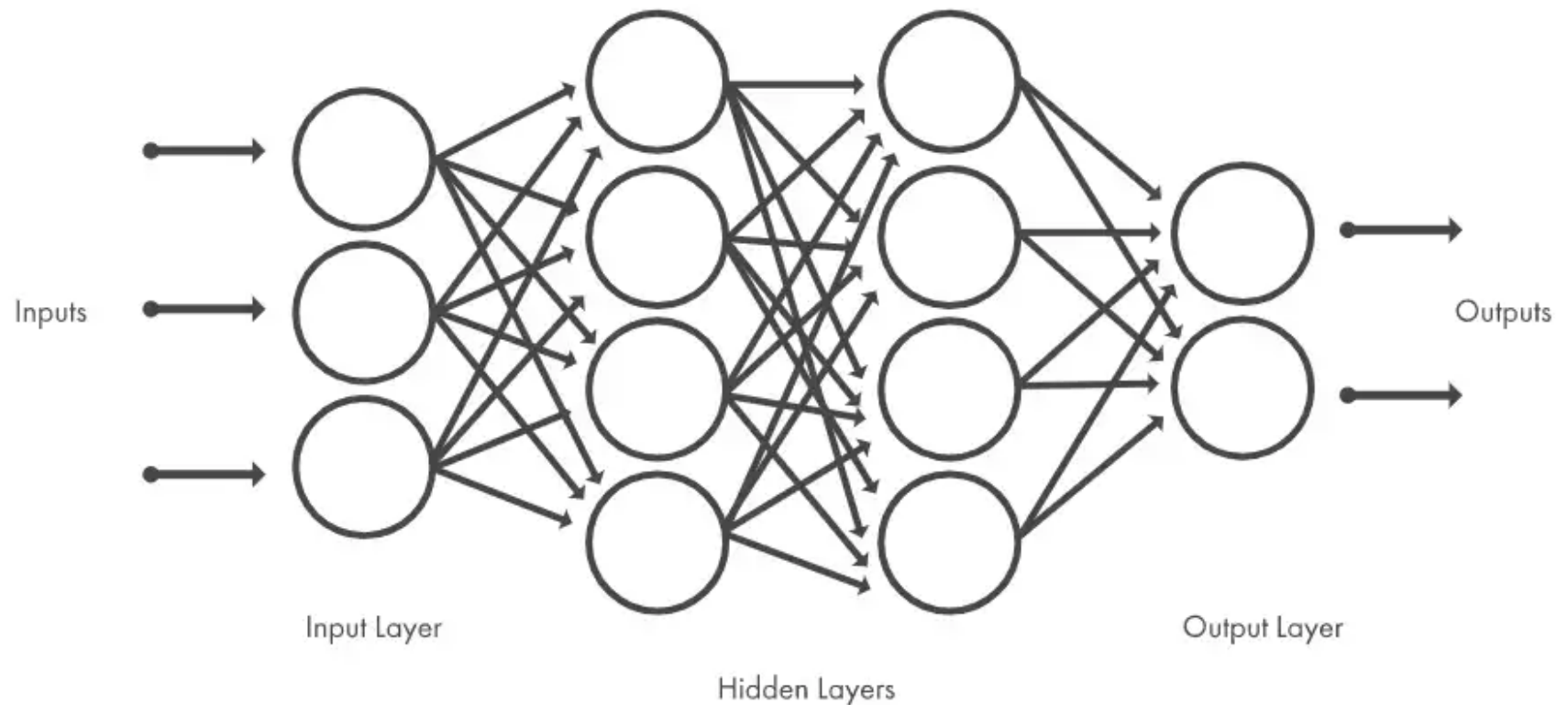




For visual types — go watch our video explaining the technicalities of image recognition

What makes a neural network deep? The number of hidden layers: While traditional neural networks have up to three hidden layers, deep networks may contain hundreds of them.





The architecture of a neural network, each layer consists of nodes. The number of hidden layers is optional. Source: [MathWorks](#)

How neural networks learn to recognize patterns

How do we understand whether a person passing by on the street is an acquaintance or a stranger (complications like short-sightedness aren't included)? We look at them, subconsciously analyze their appearance, and if some inherent features – face shape, eye color, hairstyle, body type, gait, or even fashion choices – match with a specific person we know, we recognize this individual. This brainwork takes just a moment.

So, to be able to recognize faces, a system must learn their features first. It must be trained to predict whether an object is X or Z. Deep learning models learn these characteristics in a different way from machine learning (ML) models. That's why model training approaches are different as well.

Training deep learning models (such as neural networks)



To build an ML model that can, for instance, [predict customer churn](#), data scientists must specify what input features (problem properties) the model will consider in predicting a result. That may be a customer's education, income, lifecycle stage, product features, or modules used, number of interactions with customer support and their outcomes. The process of constructing features using domain knowledge is called *feature engineering*.

If we were to train a deep learning model to see the difference between a dog and a cat using feature engineering... Well, imagine gathering characteristics of billions of cats and dogs that live on this planet. We can't construct accurate features that will work for each possible image while considering such complications as viewpoint-dependent object variability, background clutter, lighting conditions, or image deformation. There should be another approach, and it exists thanks to the nature of neural networks.

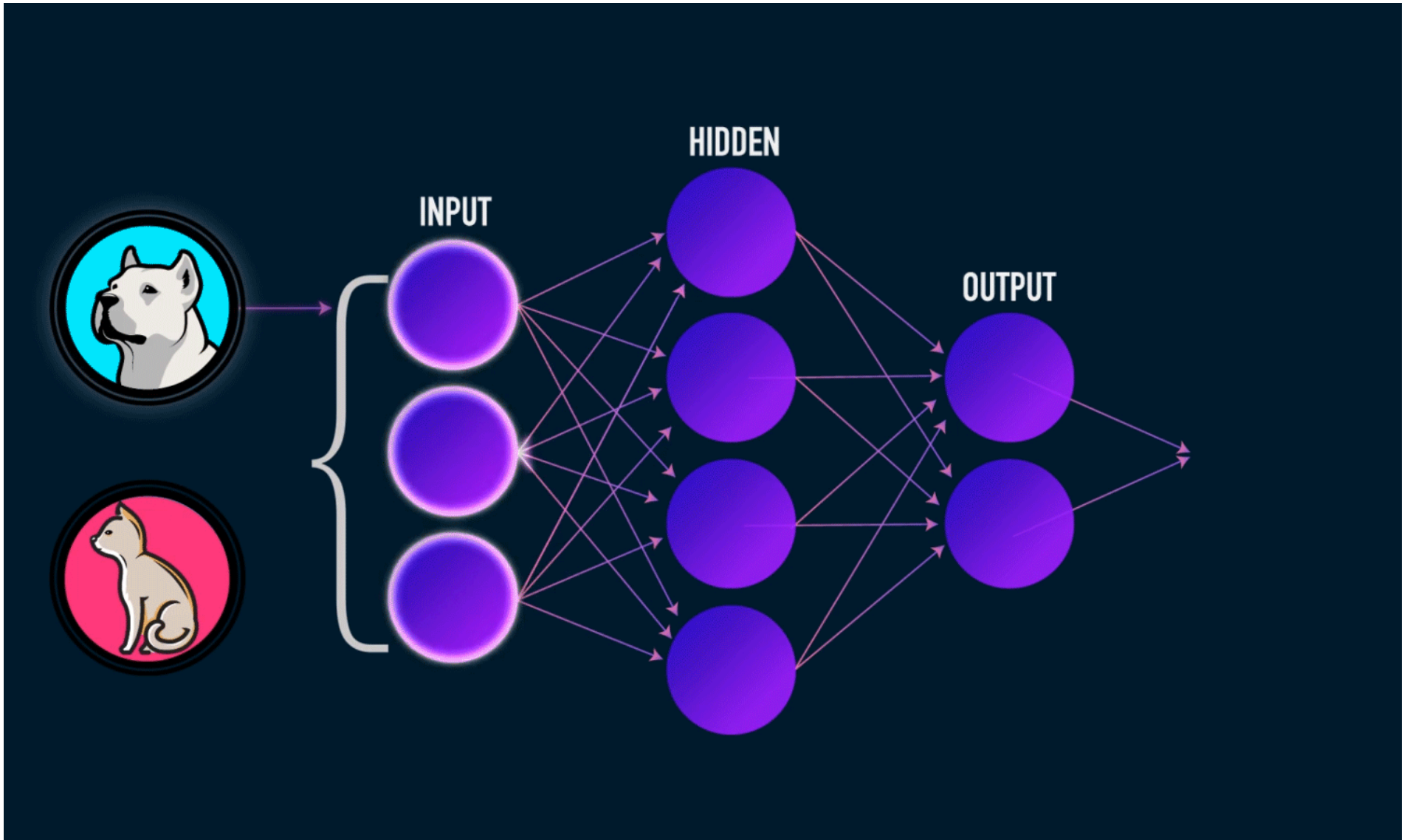
Neural networks learn features directly from data with which they are trained, so specialists don't need to extract features manually.

"The power of neural networks comes from their ability to learn the representation in your training data and how to best relate it to the output variable that you want to predict. In this sense, neural networks learn mapping. Mathematically, they are capable of learning any mapping function and have been proven to be universal approximation algorithms," notes [Jason Brownlee](#) in [Crash Course On Multi-Layer Perceptron Neural Networks](#).

The training data, in this case, is a large dataset that contains many examples of each image class. When we say a large dataset, we really mean it. For instance, the [ImageNet](#) dataset contains more than 14 million human-annotated images representing 21,841 concepts (synonym sets or synsets according to the WordNet hierarchy), with 1,000 images per concept on average.

Each image is annotated (labeled) with a category it belongs to – a cat or dog. The algorithm explores these examples, learns about the visual characteristics of each category, and eventually learns how to recognize each image class. This model training style is called *supervised learning*.

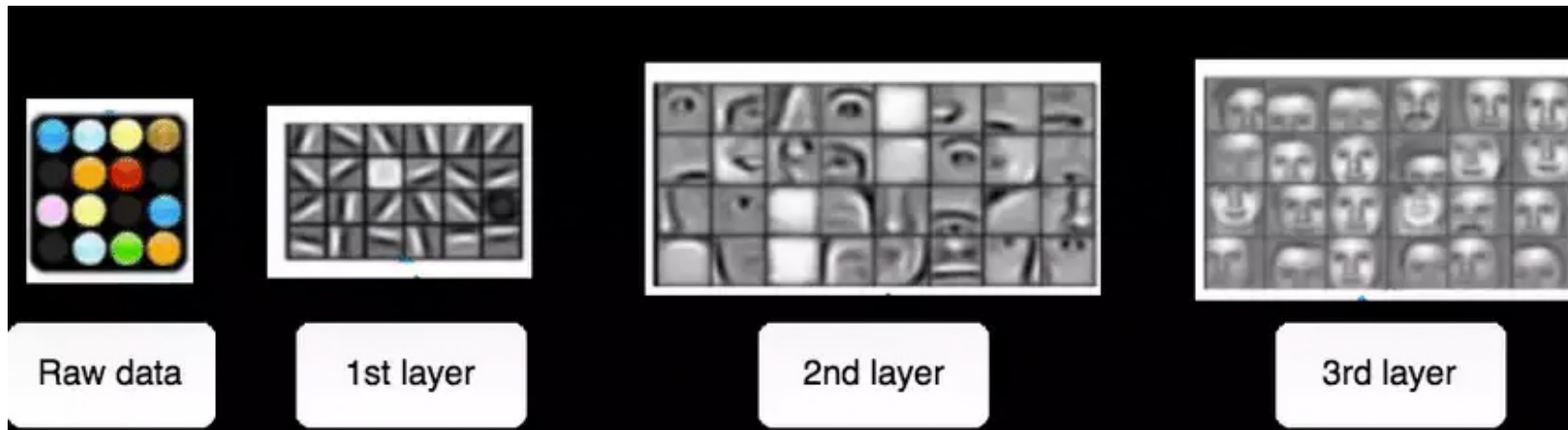




The illustration of how a neural network recognizes a dog in an image. Source: [TowardsDataScience](#)

Each layer of nodes trains on the output (feature set) produced by the previous layer. So, nodes in each successive layer can recognize more complex, detailed features – visual representations of what the image depicts. Such a “[hierarchy](#) of increasing complexity and abstraction” is known as **feature hierarchy**.





The Example of feature hierarchy learned by a deep learning model on faces from Lee et al. (2009). Source: [ResearchGate.net](#)

So, the more layers the network has, the greater its predictive capability.

The leading architecture used for image recognition and detection tasks is Convolutional Neural Networks (CNNs). Convolutional neural networks consist of several layers with small neuron collections, each of them perceiving small parts of an image. The results from all the collections in a layer partially overlap in a way to create the entire image representation. The layer below then repeats this process on the new image representation, allowing the system to learn about the image composition.

The history of deep CNNs dates back to the early 1980s. But only in the 2010s have researchers managed to achieve high accuracy in solving image recognition tasks with deep convolutional neural networks. How? They started to train and deploy CNNs using graphics processing units (GPUs) that significantly [accelerate](#) complex neural network-based systems. The amount of training data – photos or videos – also increased because mobile phone cameras and digital cameras started developing fast and became affordable.

Use cases of image recognition

Now you know about image recognition and other computer vision tasks, as well as how neural networks learn to assign labels to an image or multiple objects in an image. Let's discuss a few real-life applications of this technology.

Logo detection in social media analytics



Brands monitor social media text posts with their brand mentions to learn how consumers perceive, evaluate, interact with their brand, as well as what they say about it and why. That's called social listening. The type of social listening that focuses on monitoring visual-based conversations is called (drumroll, please)... visual listening.

The fact that more than 80 percent of images on social media with a brand logo do not have a company name in a caption complicates visual listening. How to gain insights into this case? With logo detection.

Meerkat startup conducted an experiment to show how logo detection can aid visual listening. During the six months, startupper were collecting tweets with words commonly used in the context of beer, for instance, *beer*, *cerveza*, *barbecue*, *bar*, and others. They trained a system to detect logos of popular beer brands: Heineken, Budweiser, Corona, Bud Light, Guinness, and Stella Artois. And they used it to analyze images from the tweets containing brand logos.





Heineken logo in different contexts. Source: [Meerkat's Medium](#)

Specialists indexed tweet metadata to gain insights about each brand's market share and its consumers.

First, they compared the number of posts with logos of each brand with their market share and found out that these two parameters aren't interrelated. Next, specialists extracted geo-coordinates for nearly 73 percent of tweeted images to evaluate brand presence across regions. They then plotted the percentage of each beer for the top five countries in the dataset. For instance, Bud Light is the most popular in the US, while

Heineken has fans in various countries with the biggest shares in the US and UK. The team also analyzed images that contained faces to detect the gender of beer drinkers. The difference was minor: 1.34 percent more men posted the pictures.

It's not only measuring brand awareness. Businesses are using logo detection to calculate ROI from sponsoring sports events or to define whether their logo was misused.

Medical image analysis

Software powered by deep learning models help radiologists deal with a huge workload of interpreting various medical images: computed tomography (CT) and ultrasound scans, magnetic resonance imaging (MRI), or x-rays. IBM stresses that an emergency room radiologist must examine as many as 200 cases every day. Besides that, some medical studies contain up to 3,000 images. No wonder that medical images account for nearly 90 percent of all medical data.

AI-based radiology tools don't replace clinicians but support their decision-making. They flag acute abnormalities, identify high-risk patients or those needing urgent treatment so that radiologists can prioritize their worklists.

IBM Research division in Haifa, Israel, is working on Cognitive Radiology Assistant for medical image analysis. The system analyzes medical images and then combines this insight with information from the patient's medical records, and presents findings that radiologists can take into account when planning treatment.

IBM researchers bring AI to radiology





Demo for IBM's Eyes of Watson breast cancer detection tool that uses computer vision and ML. Source: [IBM Research](#)

Scientists from this division also developed a specialized deep neural network to flag abnormal and potentially cancerous breast tissue.

[Aidoc](#) provides another solution that uses deep learning for scanning medical images (CT scans particularly) and prioritizing patient lists. The solution received clearances from the US Food and Drug Administration (FDA), Therapeutic Goods of Australia (TGA), and European Union CE markings for flagging three life-threatening conditions: pulmonary embolism, cervical-spine fracture, and intracranial hemorrhage.

The company clients include UMass Memorial Medical Center in Worcester, Massachusetts, Montefiore Nyack Hospital in Rockland County, NY, and Global Diagnostics Australia, an imaging center.

Apps for recognizing artworks

[Magnus](#) is an image recognition-fueled app that guides art lovers and collectors “through the art jungle.” Once a user takes a photo of a piece of art, the app provides such details as author, title, year of creation, dimensions, material, and, most importantly, current and historic price. The app also has a map with galleries, museums, and auctions, as well as currently showcased artworks.

Magnus sources information from a database of over [10 million images](#) of artworks; information about pieces and prices is crowdsourced. Interesting fact: Leonardo DiCaprio invested in the app, Magnus [says](#) on its Apple Store page.



Museumgoers can satisfy their hunger for knowledge with apps like [Smartify](#). Smartify is a museum guide you can use in dozens of the world's well-known art spots like The Metropolitan Museum of Art in New York, Smithsonian National Portrait Gallery in Washington DC, the Louvre in Paris, Amsterdam's Rijksmuseum, the Royal Academy of Arts in London, The State Hermitage Museum in Saint Petersburg, and others.

Smartify - Now with Augmented Reality



How Smartify works. Source: [Smartify](#)



To reveal details about a piece of art, the app matches scanned artworks against digital images in a database, which contained nearly 50,000 art pieces as of 2017. Smartify co-founder Anna Lowe [explains](#) how the app works this way: *"We scan artworks using photos or digital images and then create digital fingerprints of the artwork, meaning that it is reduced to a set of digital dots and lines."*

Facial recognition to improve airport experience

Facial recognition is becoming mainstream among airlines that use it to enhance boarding and check-in. There are two main directions of these upgrades: to follow the trends for self-service and this biometric technology and make the airport experience safer and faster. The fewer steps both passengers and staff must make to proceed with pre-flight routines, the better.

Boarding equipment scans travelers' faces and matches them with photos stored in border control agency databases (i.e., U.S. Customs and Border Protection) to verify their identity and flight data. These could be photos from IDs, visas, or other documents.

American Airlines, for instance, started using facial recognition at the boarding gates of Terminal D at Dallas/Fort Worth International Airport, Texas. Instead of using boarding passes, travelers get their face scanned. The only thing that hasn't changed is that one must still have a passport and a ticket to go through a security check. Biometric boarding works on an opt-in basis.

Facial recognition being used at DFW Airport to board American Airlines international flights



Biometric boarding for American Airlines passengers. Source: [The Dallas Morning News](#)

In 2018, American was testing biometrics for 90 days at Los Angeles International Airport Terminal 4 with the idea of expanding the use of technology if the trial goes well.

Numerous airlines implement facial recognition as an extra boarding option as well: JetBlue, British Airways, AirAsia, Lufthansa, or Delta. The last one installed a [self-service bag drop](#) at the Minneapolis-St. Paul International Airport in 2017.

Visual product search

Boundaries between online and offline shopping have disappeared since visual search entered the game. For instance, the Urban Outfitters app has a Scan + Shop feature, thanks to which consumers can scan an item they find in a physical store or printed in a magazine, get its detailed description, and instantly order it. Visual search also enhances the online shopping experience.

Apps with this capability are powered by neural networks. NNs process images uploaded by users and generate image descriptions (tags), for instance, garment type, fabric, style, color. Image descriptions are matched against items in stock together with their corresponding tags. Search results are presented based on a similarity score.

We dedicated a section about visual search in the article about [how retailers use AI](#). There you can also read about how image and facial recognition technologies have turned cashierless stores like Amazon Go into a reality and also how they power surveillance systems or enable in-store personalization.

The work goes on



In the second part of the 20th century, researchers estimated it would take a relatively short amount of time to solve a computer vision problem, among other things. In 1966, mathematician, and former co-director of MIT Computer Science & AI Lab Seymour Papert was coordinating the [Summer Vision Project](#). The researchers had an ambitious plan: [to build](#) a significant part of a system with computer vision capabilities, as we know them today, during one summer. *"The primary goal of the project is to construct a system of programs which will divide a vidisector picture into regions such as likely objects, likely background areas, and chaos,"* the project description said.

Well, it took much longer. Modern software can recognize a large number of everyday objects, human faces, printed and handwritten text in images, and other entities (check out our article on [image recognition APIs](#).) But the work goes on, and we'll continue witnessing how more and more businesses and organizations implement image recognition and other computer vision tasks to stand out from competitors and optimize operations.

Subscribe to our newsletter

[Subscribe](#)

☐ Yes, I understand and agree to the [Privacy Policy](#)

[Add a comment](#) >

Further Reading





Image Recognition APIs: Google, Amazon, IBM, Microsoft, and more



Comparing Machine Learning as a Service: Amazon, Microsoft Azure, Google Cloud AI, IBM Watson



How to Choose a Data Science and AI Consulting Company

